2001年10月1日発行（毎月1回1日発行）第20巻第10号 昭和57年12月6日第三種郵便物認可

マイコン

わかる！動かせる！
プログラムが組める雑誌

BASIC Magazine

OCT.2001 10



定価 550Yen



ポリゴンモデルを使ったシューティング・ゲーム

プレステ2でプログラミング！ 完結編

マンガで学ぼう！
パソコンレクチャー
2進数を理解する

エレクトロニック研究所訪問記
神奈川工科大学
電気電子工学科／福祉システム工学科

●自作一代記
「磁界検知式鉱石ラジオ」製作

ロボ道場番外編



ASIMO／P3／レゴが競演
ASIMO WORLD 2001 in もてぎ完全レポート

プログラマー養成講座

Technical News Spot
マイクロソフトVSボーランドのプログラム開発環境，今年はどちらが勝つ？
はじめようMicrosoft ExcelでのBASICプログラミング
toto予想プログラムにチャレンジ（2）
図解！Visual Basicゲーム作成術
キャラクター・アクション・ゲーム① フィールド上のキャラクター移動
DirectXテクノロジーを伝授する！ IE+DirectAnimation
マウスのイベント処理テクニック
携帯電話で動くゲームを作ろう！！
iアプリでロールプレイング・ゲームに挑戦
ワンダースワンのソフトを自作しよう！ チャレンジ！ワンダーウィッチ
ライン描画に挑戦

読者が作った作品をドーンと紹介
特選投稿プログラム コーナー

VisualBasic6.0	くるっと回転
VisualBasic6.0	エジプトの橋
VisualBasic6.0	ウニのさわ君2001
VisualBasic6.0+DirectX	（超大作）迷路
VisualC++6.0	（実用）BCBCクラス&サンプル
VisualC++6.0	（超大作）必殺ぷれいや～
VisualC++6.0	（超大作）P.P.Year
JavaScript	モグラ叩き in Java Script
iアプリ	牛丼食えスト外伝　牛丼道
iアプリ	（実用）iedit
WonderWitch	MIRRORS-FRASH
FX-890P	雨降りPakupa

●CMM試聴室

MU80	DRIVE
VSC-88	ビー玉の瞳











専用ホームページ
http://www.basicmagazine.net/

アルファのブロードバンド
フレッツ・ADSL（コース）使い放題、

毎日24時間使っても
月々わずか980円!!



アルファで得した分、おこづかい上げてよね。

お申込はオンラインが簡単！

24時間 オンライン申込受付中
www.alpha-net.ne.jp

24時間 資料請求受付電話（音声応答システム）
TEL:03-4306-0789

24時間 e-mail資料請求
info@alpha-net.ne.jp

24時間 FAX情報サービス（お申込書が入手できます）
TEL:03-3376-9898

今なら オリジナルストラップを入会者全員にプレゼント！



8/1〜10/31まで
期間中、アルファインターネット入会者にもれなくアルファ・オリジナルストラップをプレゼントします。
（全てのコース入会者が対象で月払い、年払いに関係なくプレゼント）

つなぎっぱなしの方には断然お得な専用コース！
入会金無料！ フレッツ・ISDNコース
月額300円（年額3,300円）
●常時接続 ●完全定額制（NTT料金別） ●HP容量15MB ●メールアドレス2つ無料（1つは弊社からの案内を含む受信専用アドレス）
※工事費1000円（初回のみ）

高速接続を重視される方にはぴったりの専用コース！
入会金無料！ フレッツ・ADSLコース（ブロードバンド）
月額980円（年額10,780円）
●高速・大容量常時接続 ●完全定額制（NTT料金別） ●HP容量15MB ●メールアドレス2つ無料（1つは弊社からの案内を含む受信専用アドレス）
※工事費1000円（初回のみ）

基本コースにプラス0円でISDNが使える充実コース！
入会金無料！ アルファ基本コース＋フレッツ・ISDNオプション（+200円でエアーエッジOK）
月額980円（年額9,800円）
●無制限・使い放題（電話代別） ●HP容量15MB ●メールアドレス2つ無料（1つは弊社からの案内を含む受信専用アドレス） ●NTT、KDDI、各種携帯・PHS、モバイル使用可能 ●常時接続、完全定額制（NTT料金別）
※工事費1000円（初回のみ）

自宅では高速接続、外出先からも使える万能コース！
入会金無料！ アルファ基本コース＋フレッツ・ADSLオプション（ブロードバンド）（+200円でエアーエッジOK）
月額1,880円（年額20,680円）
●無制限・使い放題（電話代別） ●HP容量15MB ●メールアドレス2つ無料（1つは弊社からの案内を含む受信専用アドレス） ●NTT、KDDI、各種携帯・PHS、モバイル使用可能 ●高速・大容量常時接続、完全定額制（NTT料金別）
※工事費1000円（初回のみ）

いつでもどこでも使えるダイヤルアップの定番コース！
入会金無料！ アルファ基本コース（+200円でエアーエッジOK）
月額980円（年額9,800円）
●無制限・使い放題（電話代別） ●HP容量15MB ●メールアドレス2つ無料（1つは弊社からの案内を含む受信専用アドレス） ●NTT、KDDI、各種携帯・PHS、モバイル使用可能

DDIポケットのAirH"で本格モバイル体験！
入会金無料！ エアーエッジコース *New!*
月額980円（年額10,780円）
●HP容量15MB ●メールアドレス2つ無料（1つは弊社からの案内を含む受信専用アドレス） ●無制限・モバイル常時接続（DDIポケット料金別）
※工事費1000円（初回のみ）

●アルファ・ドメインコース以外のコースではホームページの商用利用は出来ません。●受信専用メールアドレスは後日Eメールにてお送りします。（アルファ・ドメインコースを除く） ●エアーエッジオプションをお申込の方は初回のみエアーエッジ工事費1000円が別途必要となります。 ※金額は全て税別です。

NTT各種サービスも同時にお申込できます。

「フレッツ・ISDN」「フレッツ・ADSL」のアクセスポイント
全国47都道府県にサービス提供中！
※一部市町村にご利用いただけない場合があります。

ダイヤルアップ全国268ヵ所アクセスポイント

詳しくはFAX情報サービス、Webサイトをご利用ください

［AirH" 2001年内 ご入会特典］
●当社よりAirH"Card（SII製MC-P300）を購入された方は、エアーエッジコース料金を年内無料！ アルファ基本コースご入会時にエアーエッジオプションを申し込まれた方は同オプション料金を1年間無料！（初年度のみ）

全国出張サポート受付中（有料）
アルファインターネットの接続から基本講習まで、全国に出張サポートいたします。詳しくはTEL.03-3377-6655までお電話ください。

あなたに、プラス
αinternet®
プロバイダ6年の実績が支える信頼のインターネットサービスを提供します。

マイコン BASIC Magazine

CONTENTS

10 OCT.2001

専用ホームページ http://www.basicmagazine.net/

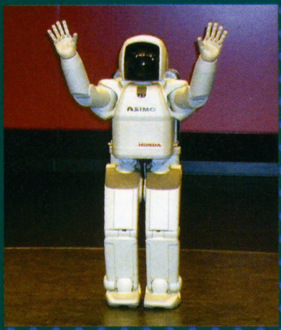
広告さくいん

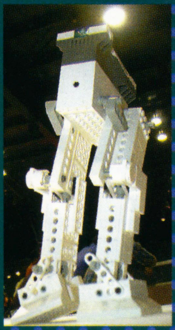
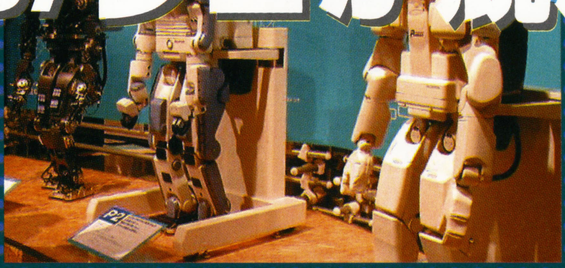
第20巻 第10号 2001年9月25日印刷
2001年10月1日発行
定価550円（本体524円）
電波新聞社　2001 Printed in Japan
編集発行人：平山哲雄
東京都品川区東五反田1-11-15
郵便番号141-8715
電話　東京(03)3445-6111（大代表）
振替　（東京）00150-3-51961
印刷：奥村印刷(株)　製本：(株)堅省堂

21世紀の遊びはコレ！ ロボ道場 番外編

2足歩行ロボット

ASIMO/P3/レゴが競演



今月は，番外編として，「ASIMO WORLD 2001 in もてぎ」を完全レポートします。じつは，ベーマガ編集部もこのイベントの企画段階に加わったんですよ。　（松原拓也）

ASIMO WORLD 2001 in もてぎ

本田技研工業
http://www.honda.co.jp/

Hondaファンファンラボ
http://www.honda.co.jp/fanfunlab/

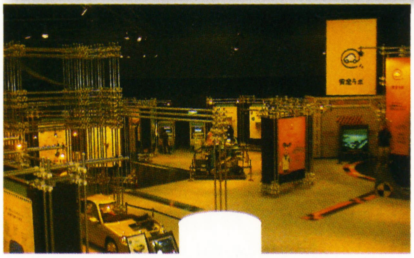
ツインリンクもてぎ
（〒321-3597 栃木県芳賀郡茂木町大字桧山120番地1）
http://www.twinring.co.jp/



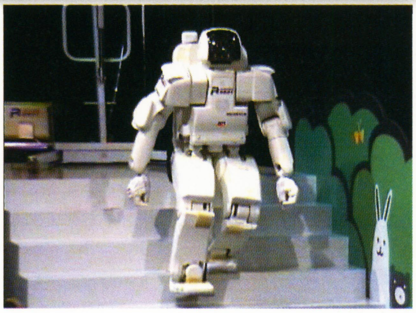
8月4～14日までの11日間（8月7日のみ休館日），栃木県の「ファンファンラボ（ツインリンクもてぎ内）」で「ASIMO WORLD 2001 in もてぎ」というイベントが行われました。内容は，自律二足歩行ロボット「ASIMO」や「P3」の歩行デモンストレーションの他に，ロボットのおもちゃを集めたコーナー（ロボトモひろば）や，ASIMOを題材にしたコンピュータ・ゲーム，そして「CAM-08」や「レゴ·マインドストーム」を使った工作教室など，盛りだくさんな内容でした。


▲ステージは円形で2階からの見物もできる。最終日は大勢のお客で賑わった

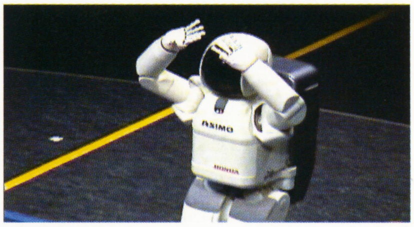
ASIMOといっしょ！オンステージ P3とASIMOが夢の競演!!

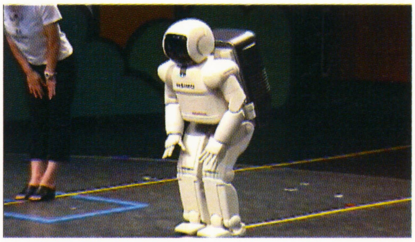

▲館内には電気自動車やドライブ・シミュレータ，ジェット機などの最新技術が展示されている


▲「ドキット（二輪）」と「ワクッチョ（四輪）」の組み立てコーナー（2001年1月号でレポート記事掲載）


▲全高160cm，重量130kgの「P3」（1997年9月完成）。階段を降りる歩行デモは圧巻


▲「ASIMOと遊ぼう」。子供達と「だるまさんが転んだ」をするASIMO（大敗）


▲全高120cm，重量43kgの「ASIMO」（2000年11月誕生）。「i-WALK」技術で自在な歩行を実現（写真は盆踊りをするASIMO）

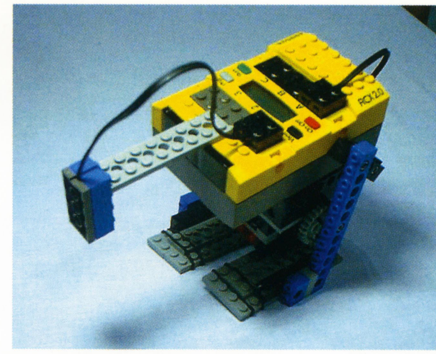

▲ASIMOの操作はステージ袖のスタッフ2名によって行われている（写真はお辞儀をするASIMO。カワイすぎ!!）

つくろ！あそぼ！ASIMO工作室
レゴブロックで作るオリジナル歩行ロボット製作教室



「ASIMO WORLD 2001 in もてぎ」では，レゴ・マインドストームを使った工作教室が行われました。教室では僕（松原）と桐林君など計4名がマインドストームの先生として参加しました。テーマはずばり「二足歩行」です。

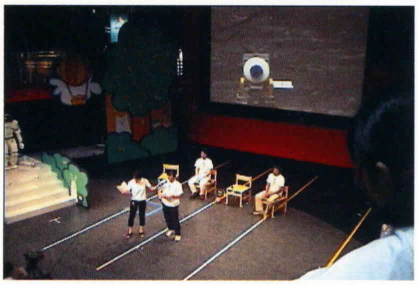
◀「オリジナル歩行ロボット製作教室」のようす



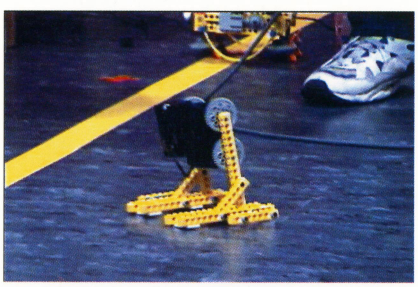
▲ASIMOにちなんで製作された二足歩行ロボット（作：松原）。ライトセンサで床を検知して，自動的に停止する

レゴ・マインドストーム・スペシャルプログラム

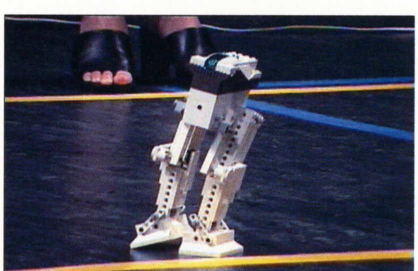
8月11～12日には「レゴ道場」でおなじみの桐林君や五十川さんなど計6名がステージでレゴ・マインドストームによる作品を発表しました。会場にASIMOとレゴ・マインドストームが同居するステージがとても画期的でした。



▲レゴ・マインドストームの作品をステージで発表



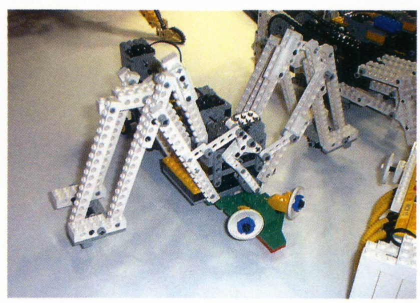
▲くるとんさんの二足歩行ロボット。機構はクランク運動だけの簡単なもの



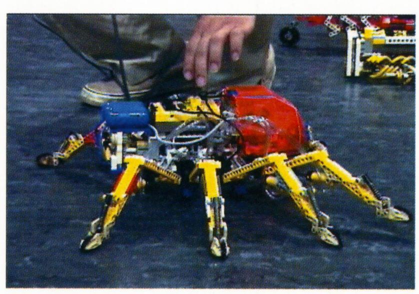
▲五十川さんの「純白の半神（改）」。ちゃんとヒザが動き，片足が上がる二足歩行ロボット



▲桐林君の「お掃除ロボット」。ホウキは左右どちらにも出せて，目玉が動く



▲かぜらさんの「メカンボウ」。両腕を開閉する力だけで移動するロボット



▲ぴこさんの超大作「タランチュラ」。本物と同じ足運びがすごい。足の持ち上げには空気圧（ニューマチック）を使用



▲桐林君の「ロボットアーム」。アームを使って，ペットボトルのジュースをコップに注ぐという荒業に挑戦して成功



▲やぐちさんの「ぺんぎん」。シンプルな構造ながら，動きはバタバタと派手。胴体にはMicroScout（※）を使用
※Droid Development Kit（DDK）に付属。あらかじめ七つのプログラムが入っている。



▲ぴこさんの「アシモ・ターボバイク」。「ASIMOレゴ」が自転車をこぐ姿がユーモラス

ロボ道場 ツインリンクもてぎ

「ツインリンクもてぎ」は栃木県茂木町にあるモビリティー&レクリエーション施設です。「ツインリンクもてぎ」内



▲日本初の本格的オーバルコース「SuperSpeedway」



▲森の探見ミュージアム「ハローウッズ」（「探見」は探検十発見の造語）

には，「SuperSpeedway（全長距離2.4km）」や「RoadCourse（全長距離フルコース4.8km）」などのコースの他に「ファンファンラボ」「ホンダコレクションホール」「ハローウッズ」「アクティブ・セーフティ・トレーニングパークもてぎ」などの施設があります（全部紹介できないのが残念）。

ロボ道場 最後に

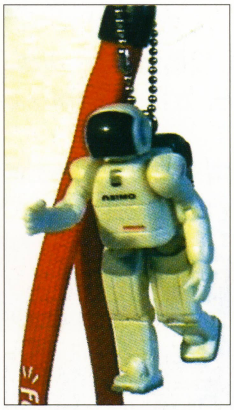
レゴ・マインドストーム（とベーマガ）のPR活動に完全燃焼した充実の11日間でした（さらにASIMOと握手ができて満足）。来年の夏にも「ASIMO WORLD 2002 in もてぎ」が開催されるそうですので，お楽しみに。



▲約450台の二輪・四輪などが動態保存されている「ホンダコレクションホール」



▲おまけ。ASIMOと握手する松原（かなり緊張）



◀ヒザが可動する特製「ASIMOストラップ」を5名様にプレゼント！（提供：Hondaファンファンラボ）。くわしくは読者プレゼントのページを見てね（80ページ）

■ロボカップトイズ（RoboCup Toys）

タカラ・トミー・バンダイの3社が「ロボカップ」の理念に賛同して「ロボカップトイズ」を創設。各社からサッカーロボットが発売されました（オフィシャル乾電池はマクセル）。



▲各パーツの固定は「ロックピン」と「ジョイントツール」を使う



▲コントローラは赤外線式。最大8台までの同時プレイが可能

パーツは着脱可能で「コアユニット」「バッテリーボックス」「メインフレーム」は全社共通，「ボディ」「走行ユニット」「キックユニット」「ホールドユニット」は各社が独自に開発しています。

2002年に福岡ドームで「ロボカップトイズ」の全国大会が予定されています。

■ロボットは3タイプ!!

3タイプはいずれも定価4,980円（税別）。コントローラとボールが付属（電池は別売）。「ツクモロボコンマガジン館」や，全国のおもちゃ屋で発売中。



▲タカラ「ジェイストライカー」
寸法：240×150×82mm
色：ブルー／ホワイト
特徴：上体を持ち上げる「リフト機構」を搭載。

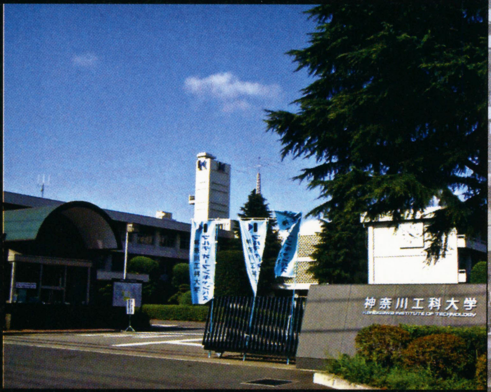


▲トミー「ブレイクリーガー」
寸法：275×152×112mm
色：スカイブルー／ファイアーレッド
特徴：ボールのホールド機能がなく，速攻性に優れたマシン。



▲バンダイ「フィールドコマンダー」
寸法：275×156×100mm
色：ブラック&レッド／ホワイト&ブルー
特徴：強力にボールを挟みこむ「クロスホーンシューター」を搭載。

＊撮影協力：
ツクモロボコンマガジン館



エレクトロニック研究所訪問

神奈川工科大学

電気電子工学科　金井研究室
福祉システム工学科　山本研究室



神奈川工科大学　http://www.kanagawa-it.ac.jp/

神奈川県厚木市にある神奈川工科大学は7万坪という広大なキャンパスを備えた大学です。その中には六つの実験棟，講義棟，図書館，学生会館，各実験実習棟など，数多くの建物があります。

大学の歴史は，昭和38年に幾徳工業高等専門学校を開校したのが始まりで，その後，昭和50年に幾徳工業大学を開学。昭和63年に大学名を神奈川工科大学に変更しました。現在は工学部(4年間)に「機械工学科」「電気電子工学科」「応用化学科」「システムデザイン工学科」「情報工学科」「福祉システム工学科」「情報ネットワーク工学工学科と，大学院・工学研究科に「機械工学専攻」「電気電子工学専攻」「応用化学専攻」「機械システム工学専攻」「情報工学専攻」の5専攻を持つに至っています。

今回は大学内にある97もの研究室の中から，「山本研究室」と「金井研究室」を紹介します。

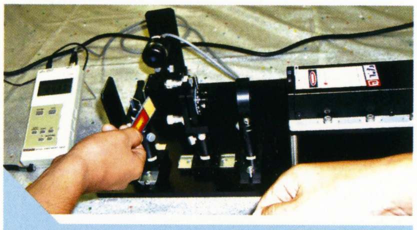
電気電子工学科　金井研究室

電気電子工学科WEBサイト　http://www.ele.kanagawa-it.ac.jp/


金井研究室の金井徳兼助教授(去年までSKY PerfecTV!で放送された「わんぱくレゴ星人」でもおなじみ

電気電子工学科では，電流と電子の働きを利用することを主題として「半導体と光を使った電子デバイス」「電子情報通信システム」「電力・制御・機器」という3分野に力を注いでいます。

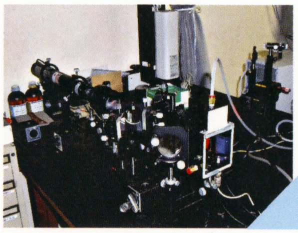

ゼミ生がレゴ・マインドストームで製作したブロックを色別に仕分ける装置。RCXを三つ使用

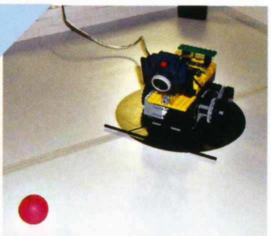

KTP結晶による非線形光学効果を応用して，その利用方法を研究中

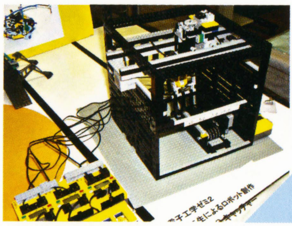
電気電子工学科の金井研究室では光計測や光物理という光を扱った研究のほかに,ロボットの研究も行っています。

ロボット関連では，マイクロメカニズムコンテスト（精密工学会主催）参加用に開発した20ミリ立方の超小型ロボットや，レゴ・マインドストームを使ったロボットを製作しています。

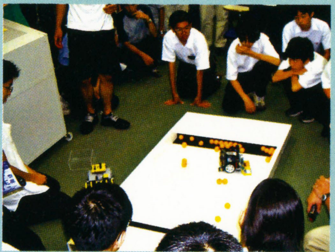
金井研究室は大学内で最もレゴ・マインドストームの普及している研究室です。その成果は，ネットワークを介したRCXの遠隔操作の研究や,「電子ロボと遊ぶアイデアコンテスト」の大会運営などでも発揮されています。


3次元画像情報の記録・再生が可能なホログラム。正面から覗き込むと像が浮かび上がる


ロボットによるスポーツ競技というテーマの一つ，「ビジョンコマンド」を使ったサッカーロボット


こちらもゼミ生による「UFOキャッチャー」風の作品。RCXを三つ使用

「第3回　電子ロボと遊ぶアイデアコンテスト」


◀コンテストの競技フィールド。ボールをシュートするロボットに全員が注目

神奈川工科大学主催によるロボット競技が「ロボフェスタ神奈川2001」の相模原大会で行われます。高校生による38チームが，レゴ・マインドストームで作ったロボットでそのアイデアを競います。
開催日：平成13年10月7日(日曜日)
会場：銀河アリーナ　神奈川県相模原市
くわしくはWEBサイトをご覧ください
http://www.pref.kanagawa.jp/osirase/robo/

福祉システム工学科　山本研究室

福祉システム工学科WEBサイト　http://www.we.kanagawa-it.ac.jp/



福祉システム工学科（および大学院機械工学専攻）山本圭治郎 教授

「福祉システム工学科」は，2年前に誕生したばかりの新しい学科です。同学科のカリキュラムは，医学，メカトロニクス，人間工学，福祉環境学など，工学と福祉を両立している点が特色で，使う人の身になった福祉機器を設計できるスペシャリストの育成に重点が置かれています。

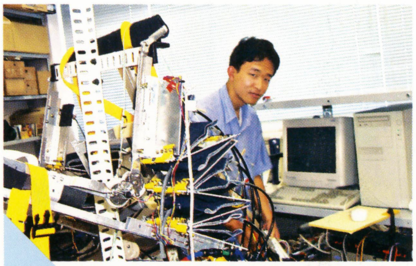
パワーアシストスーツ

山本研究室で研究開発している「パワーアシストスーツ」は，装着すると肘・腰・膝に力が補われて「力持ち」になることができます。このスーツは介護者が，体の不自由なお年寄りや患者さんを抱き起こすときの身体的な負担を軽減するための装置です。

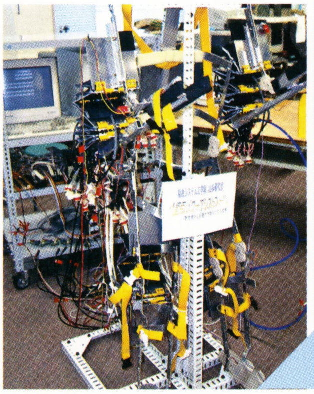
このスーツの研究が始まったのは，今から12年前。研究当初から，威圧感を与えないようにメカを体の全面に出さない点や，駆動系には空気圧を使う点が一貫しています。

関節の動作入力には5〜6年前から「筋肉硬さセンサ」という独自開発のセンサを採用しています。スーツは年1体のペースで改良され，今回の12番目の試作では，さらに軽量化と小型化が実現しました。しかし，現時点ではエアーコンプレッサやコンピュータ機器が外部にはみ出ています。最終的には，配線やバッテリ，エアーコンプレッサをスーツ内に格納し，さらにスーツの重さを10kg前後に，そしてスーツ自身の価格を20万円に抑えることを目標としています。

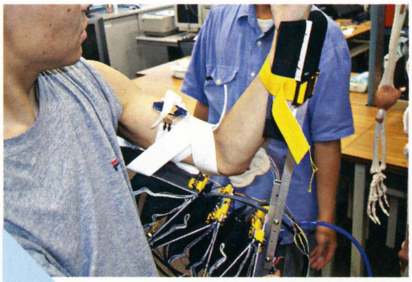
パワーアシストスーツは2003年の実用化を目指して研究中です。



大学院機械工学専攻で博士後期課程の石井峰雄さん。現在は石井さんを含め，計5名でスーツを研究中とのこと



11番目のパワーアシストスーツ試作機。まだ，コンプレッサやコンピュータが外部にはみ出ている状態



肘関節の後ろに位置する空気圧式アクチュエータ。そして，二の腕に巻いているのが「筋肉硬さセンサ」



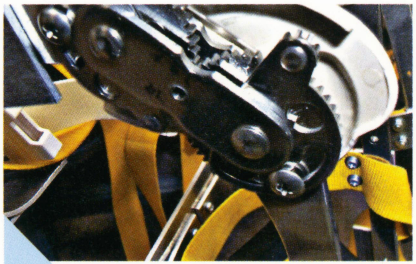
ずらりと並んだ人体模型。スーツの設計時，関節を理解するためには必要とのこと



完成したての最新型，12番目のパワーアシストスーツ試作機。堂々の小型・軽量化を実現

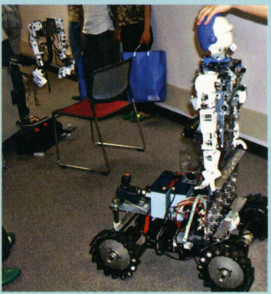


現在，アクチュエータには血圧計カフを流用。軽量・小型化のためにベローズ（蛇腹）式も研究中



人間工学に基づき，関節は装具で使われている多軸ヒンジを採用

ロボフェスタ神奈川2001に神奈川工科大学のロボットが登場!!



◀兵頭研究室の代理行動ロボット。テムザックのヒューマノイド型超遠隔操作ロボット「テムザック」を大幅に改良，機能追加をしたもの

8月から11月にわたって神奈川県内の4会場で「ロボフェスタ神奈川2001」が開催されています。

会場では，企業や大学などのさまざまなロボットなどを紹介する「ロボテックショー」というショーも行われるのですが，その中に神奈川工科大学の「パワーアシストスーツ」や「代理行動ロボット」が登場する予定です。興味のある人はぜひ会場で実物をご覧ください！

ロボフェスタ神奈川2001 開催地

[横須賀会場] 横須賀市南体育会館
会期：終了

[川崎会場] 川崎市とどろきアリーナ
会期：9月8日（土）〜9月16日（日）

[相模原会場] 相模原市銀河アリーナ
会期：10月6日（土）〜10月14日（日）

[横浜会場] パシフィコ横浜
会期：11月16日（金）〜11月25日（日）

入場料などはWEBサイトをご覧ください
http://www.pref.kanagawa.jp/osirase/robo/

わかる！おもしろい！初歩からのプログラミング講座

プレステ2でゲームをプログラムする！



完結編：ポリゴンモデルを使ったシューティング・ゲーム

「BASIC STUDIO」アシスタントプロデューサー ● アートディンク 齋藤博希

今回の内容

さて，前回の講義では，テキスト文字を使ったシューティング・ゲームを作成しました。しかし，自機も弾も敵もテキスト文字で表示されているため，画面が少々さみしいものとなっています。

今回の講義では，3Dグラフィックスの基礎としてポリゴンモデルの扱いから3D空間の考えかたを解説して，前回作成したシューティング・ゲームを3Dグラフィックスで表示されたものにバージョン・アップしてみたいと思います。

3Dグラフィックスとは

それでは始めに，ポリゴンモデルに代表される3Dグラフィックスの考えかたから学んでいきましょう。とはいってもむずかしく考えることはありません。コンピュータ上で表現される3Dグラフィックスはあなたの目に映っている風景になぞらえて説明することができます。

まず，あなたの目に映っている映像について，よく考えてみましょう。今，あなたの目には何が見えていますか？おそらく，この記事の書かれた本が見えているでしょう。さらに，本を見るには当然ですが目が必要です。また，見ることはできませんが，本や目を置くための「空間」がなければ，見える，見えない以前の問題になってしまいます。

これは，コンピュータ上で表現される3Dグラフィックスでも同じことがいえます。本のような「物体」が「ポリゴンモデル」として存在し，ポリゴンモデルを見るための「目」として「カメラ」が存在します。また，これらのポリゴンやカメラを置くための「空間」として「ワールド空間」が存在します（**第1表**）。

まあ，現実世界での見えかたをコンピュータが真似しているわけですから，よく似ているのはある意味当たり前なのですが…。

このように「見る対象」と「見るためのもの」，「これらを配置する空間」を使って，初めて3Dグラフィックスが表示されることになります。あとはモデルやカメラを動かしたりすることで，多彩な表現をすることができるのです

モデル，カメラ

モデルは形を持った「物体」にあたり，カメラは人間の「目」にあたります。それぞれワールド空間で「位置」と「向き」を指定することができます。モデルがどのようにカメラに写るかは，モデルとカメラとの相対的な位置で決まります。ですからカメラがモデルと反対方向を向いていると，当然カメラにモデルは写らないことになります。このあたりは現実世界と同じです。

PS2でプログラムするために必要なもの

PlayStation 2本体（35,000円）
BASIC STUDIO パワフルゲーム工房（9,800円）
メモリーカード（3,500円）
USBキーボード（アートディンクの通販で3,480円）
USBマウス（3,000円程度）

ワールド空間

モデルやカメラについては現実にあるものなので多くを説明しなくてもわかると思います。しかし，ワールド空間については，漠然としかイメージできないと思いますので，くわしく説明したいと思います。

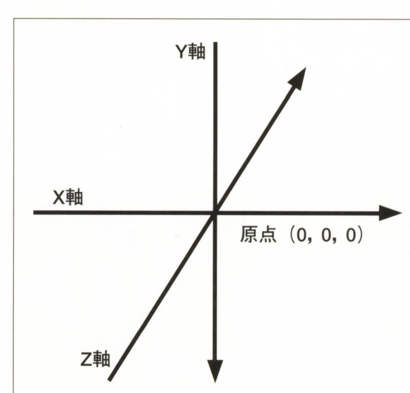
ワールド空間とは先ほども述べたとおり，モデルやカメラを配置するための空間です。しかし，ただ漠然とした空間ではモデルやカメラを配置しようとしても「どこに」置けばいいのかわかりません。

このため，ワールド空間でも前々回で説明した「座標」という考えかたを使っています。ワールド空間のある任意の場所に原点を設定し，現実世界の「縦，横，高さ」と同じような座標軸が設定されているのです（**第1図**）。

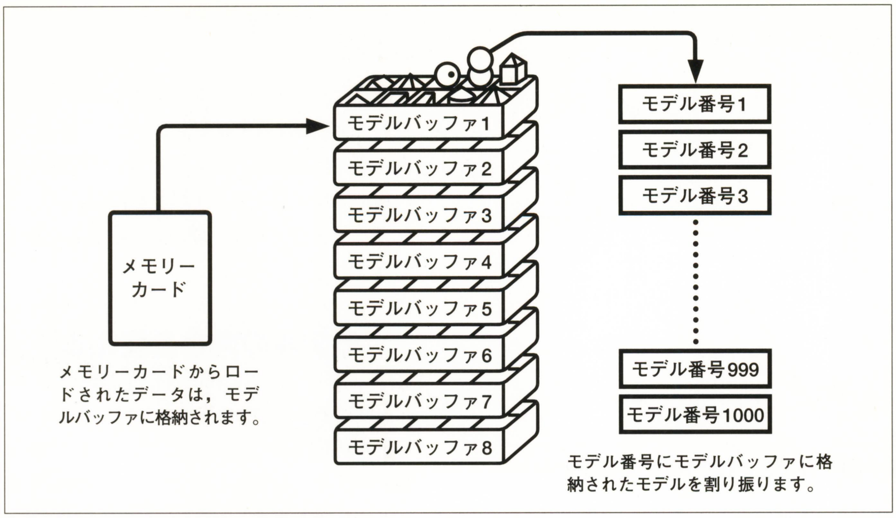
なお，**第1図**のようにワールド空間では原

《第1表》物を見るために必要なもの

現実世界	コンピュータ上	役割
物体	モデル	目やカメラに写るもの。これがないと目の前には見えない「空間」しかないので，何も写らないことになります。
目	カメラ	物やモデルを見て，映像として認識させるために必要。なければ，物があっても「見えない」
空間	ワールド空間	物や目が存在するために必要な「置き場」。置き場がなければ当然見えるものも見るものも存在しません。



《第1図》ワールド空間の座標軸



《第2図》モデルを表示させる考えかた

点（0，0，0）からX軸，Y軸，Z軸とよばれる「基準となる軸」が伸びており，それぞれが垂直に交わる関係を持っています。テキスト座標と同じようにワールド空間上で位置を示すには「X軸の方向に〜，Y軸の方向に〜，Z軸の方向に〜」といった形で表します（X座標，Y座標，Z座標）。これを「ワールド空間座標」と呼びます。なお，1文字分の大きさが単位の基準となっていたテキスト座標とちがって，ワールド空間は仮想空間のため基準となる最小単位がありません。

モデルを表示してみよう

さて，3Dグラフィックスの基本的な考えかたがわかったところで，実際にBASIC STUDIOを使ってモデルを表示してみましょう。モデルを表示させるのは非常に簡単です。

```
MODDISP 1
```

どうでしょう，画面上に4色にわけられた四角形が表示されたと思います。真正面から見ているのでわかりづらいですが，これは正八面体です。

この命令文では「MODDISP」が「モデルを表示状態にしろ」という命令で，命令の後に書かれたパラメータ「1」はモデルを管理するための番号で「モデル番号」と呼ばれます。よって，この命令文は「モデル番号1番のモデルを表示状態にしろ」といった内容になります。

つぎにこの命令文を実行してみます。

```
MODOFF 1
```

先ほど表示されていた正八面体が消え，元の真っ暗な画面に戻ったと思います。「MODOFF」は「モデルの表示状態を解除しろ」といった命令なので，この命令文によってモデル番号1のモデルの表示状態が解除され，画面上から見えなくなったことになります。このようにモデルの表示状態と非表示状態の切り替えはそれぞれ命令一つで行うことができます。

なお，“先ほどモデル，カメラ”の項で述べたとおり，モデルが表示状態になっても，カメラがちがう方向を向いていると画面上にモデルは表示されません。MODDISP命令はあくまでモデルを表示状態にするだけで，実行したからといって必ずしも画面上に表示されるわけではないことに注意してください。

モデルを表示させるまでの手順

先ほどの例では正八面体が表示されるだけで，自分の思いどおりの形は表示されません。そこで順序が少し逆になってしまいましたが，モデルを表示させるまでの手順を説明して，思いどおりの形を表示するまでを行ってみたいと思います。

思いどおりのモデルを表示するまでの手順はこのようになります。

①「POLY-CRAFT」を使ってモデルを作成する。
②作ったモデルをメモリーカードにセーブする。
③「EZ-BASIC」で作ったモデルを読み込み，「モデルバッファ」と呼ばれる置き場に置く
④1〜1000まであるモデル番号に「モデルバッファ」にある「モデル」を割り当てる
⑤モデルを表示状態にする

《第2表》初期モデル一覧表

モデルバッファ1		モデルバッファ5	
1	菱箱	1	黄ビーム
2	四角垂	2	青エネルギー球
3	円柱	3	鉄球
4	三角柱	4	隕石
5	赤三角柱	5	ミサイル
6	緑三角柱	6	火柱
7	青三角柱	7	水柱
8	黒三角柱	8	衝撃波
9	球体	9	爆発球
10	黒球	10	煙幕球

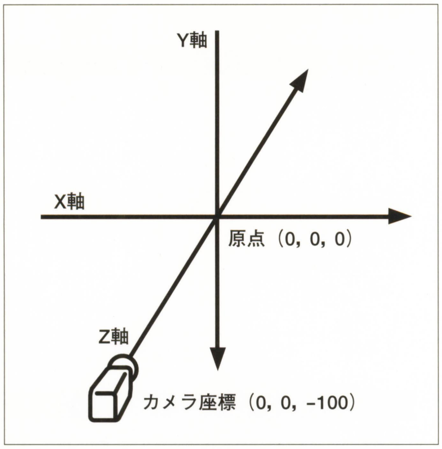
モデルバッファ2		モデルバッファ6	
1	箱	1	SD兵士
2	赤箱	2	SD魔術師
3	緑箱	3	SD僧侶
4	青箱	4	SD女王
5	黒箱	5	SD王様
6	赤ワイヤーボックス	6	SDスケルトン
7	青ワイヤーボックス	7	SDゴブリン
8	緑取り箱	8	SDサイクロプス
9	緑取り箱	9	SDデーモン
10	緑取り箱	10	SDドラゴン

モデルバッファ3		モデルバッファ7	
1	荒地箱	1	SD戦車
2	草地箱	2	SDトラック
3	砂漠箱	3	SD砲台
4	水面箱	4	SD攻撃ヘリ
5	石壁箱	5	SD戦闘機
6	木	6	SD潜水艦
7	ブッシュ	7	SD空母
8	岩石	8	SD戦艦
9	石柱	9	SD宇宙戦闘機
10	木箱	10	SDロボット

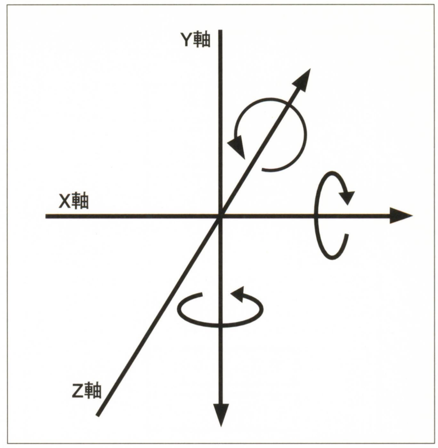
モデルバッファ4		モデルバッファ8	
1	宝箱	1	SDイヌ
2	ドラム缶	2	SDネコ
3	救急箱	3	SDウサギ
4	ハート	4	SDサル
5	薬ビン	5	SDペンギン
6	ドル袋	6	SDクジラ
7	爆弾	7	SDイルカ
8	クリスタル	8	SD UFO
9	？	9	SD火星人
10	空	10	SDグレイ

第2図にモデル表示の流れを示します。モデルバッファとはメモリーカードに保存されたモデルファイルを格納することのできるモデル置き場で，EZ-BASICでは八つのモデルバッファが用意されています。よって，モデルが保存されたファイルを同時に八つまで読み込んでおくことができます。また，モデルファイル一つにつき，最大10個までのモデルが格納できるので，同時に80種類のモデルを同時に扱うことができます。

今回はプログラミング講座なので，①，②，③のモデルの作成とメモリーカードへの保存，読み込みの手順は省きます。幸い，EZ-BASICを起動した時点で基本的なモデルは



《第３図》カメラの初期位置



《第４図》X，Y，Z 軸それぞれの回転方向

あらかじめモデルバッファに読み込まれています。

今回は，このデフォルトモデルを使うことにしましょう（**第２表**）。

モデル番号にモデルを割り当てる

モデル番号にモデルを割り当てるには「MODSET」命令を使います。それでは先ほど使ったモデル番号1番に戦闘機のモデルを割り当ててみましょう。

第2表を見ると戦闘機のモデルはモデルバッファ7のオブジェクト番号5のモデルとして格納されていることがわかります。よって，このように入力します。

```
MODSET 1,7,5
```

MODSETの後に三つ，パラメータが入力されています。この三つのパラメータは左から表示用モデル番号，モデルバッファ番号，オブジェクト番号の順番で並んでいます。

また，MODSET命令は，表示用モデル番号にモデルバッファ番号とオブジェクト番号で指定したモデルを割り振るといった命令です。つまり，ここでは「モデル番号1番にモデルバッファ7のオブジェクト番号5のモデルを割り当てろ」という内容になっています。

それでは，先ほどと同じようにMODDISPを実行してみてください。どうですか？先ほど表示された正八面体の代わりに戦闘機が表示されたと思います。

モデルの位置を変更する

モデルが表示できたら，つぎはモデルを動かしてみたくなります。その前にワールド空間でモデルとカメラがどのような位置関係になっているかを知る必要があります。

先ほど表示したモデルは，初期位置としてワールド空間の原点(0,0,0)に配置されています。また，カメラは初期位置で(0,0,-100)に配置されており，原点の方向を向いています。

よって，画面上のちょうど真ん中に原点が表示され，原点を通って右方向にX軸，下方向にY軸，画面奥に向かってZ軸が通っていることになります。このため先ほど表示状態にしたモデルは，画面中央に表示されたのです（**第３図**）。

それでは，現在のカメラとモデルの位置関係がわかったところで，実際にモデルの位置を変更してみましょう。モデルの位置はMODPOS命令にワールド座標をパラメータとして使って指定します。つぎの文を実行してみてください。

```
MODPOS 1,40,0,0
```

すると，先ほどまで表示されていたモデルの位置が画面右側に変わりました。MODPOS命令の後に書かれたパラメータは，モデル番号，X座標，Y座標，Z座標でこれらの座標を変化させることで指定したモデル番号のモデルの位置を変更することができます。数値を変化させたときにどのように位置が変わるかはカメラとモデルの位置関係によってちがってきますので，**第3図**を見てよく考えてみましょう。

モデルの向きを変える

今度はモデルの向きを変更してみましょう。モデルの向きを指定するには，X軸,Y軸,Z軸のそれぞれを軸としてどのくらい回転させるかで指定します。回転させる向きは，**第4図**を見てください。

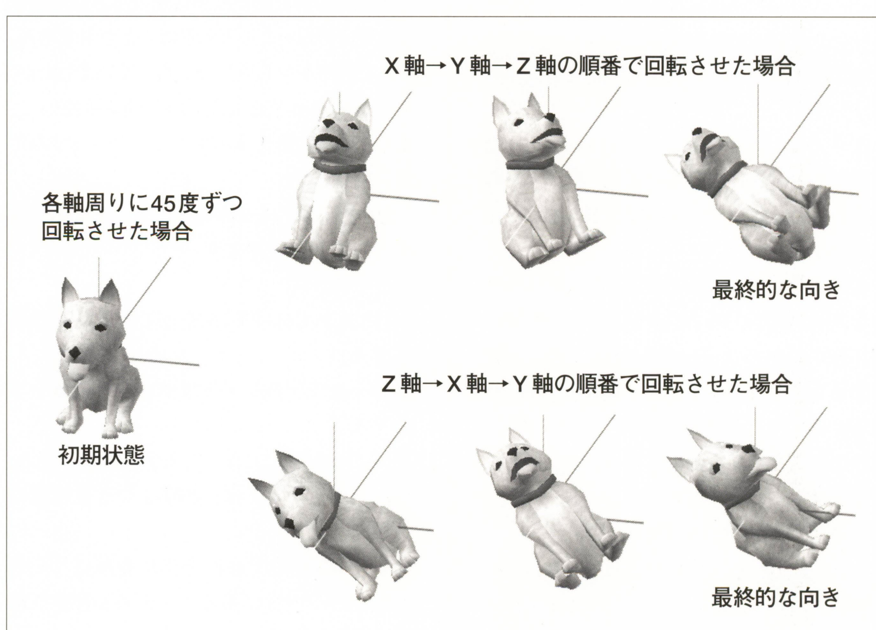
まず，つぎの命令を実行してみてください。

```
MODROT 1,-1.57,3.14,0
```

すると，戦闘機が上を向いた状態に向きが変わります。MODROT命令の後に書かれたパラメータは，モデル番号,X軸回転角,Y軸回転角,Z軸回転角となります。MODPOS命令に非常によく似ていますが，注意しなければならない点がいくつかあります。

角度の単位

一つ目のちがいは，向きを指定する際に使う回転角の単位が，私たちがよく使う「度」ではないことです。通常私たちは，1回転が360度の「度」という単位を使っていますが，EZ-BASICでは1回転が2 π（円周率のπ



《第５図》座標軸を回転させる順番によるちがい

です)の「ラジアン」という単位を使います。このため，先ほどの例では「1.57」や「3.14」といった中途半端な値を使っていたのです。

ラジアンはあまり使い慣れない単位なので，あまり扱いたくない人もいると思います。そんなときはRAD関数を使うとよいでしょう。RAD関数は引数に角度を入れることで，ラジアンに変更してくれる便利な関数です。試しにRAD 関数を使ってみましょう。

```
PRINT RAD(180)
```

すると，180度がラジアンに変更されてπとなり，その数値「3.14159」が表示されます。

RAD関数を使って先ほどのMODROT命令文を書き直すと，以下のようになります。

```
MODROT 1,RAD(-90), RAD(180), RAD(0)
```

これならば，x 軸に90 度，y 軸に180 度モデルを回転させて向きを指定していることがすぐにわかりますね。

回転させる順番

二つ目のちがいは，X軸，Y軸，Z軸を軸として回転をさせるときに順番があるということです。先ほどの位置を指定するMODPOSでは，どのパラメータから扱っても結果にちがいはありませんが，向きの指定では回転させる順番がちがうと結果までちがってきます（第5図）。

なお，EZ-BASICではZ軸→X軸→Y軸の順番で回転を行っています。

向きの指定は，このように癖があるため，実際にペンや模型などを手で動かして，確認しながら設定するとよいでしょう。

その他の設定

先ほどまでの「位置の指定」と「向きの指定」の他にも，モデルに対して拡大率などいくつか設定できるものがあります。これらの設定を行うのが「MODOPT」命令です。つぎの命令を実行してみてください。

```
MODOPT 1,2,1,1,1
```

実行すると，先ほどまで表示されていたモデルの大きさが大きくなります。MODOPT命令の後に書かれたパラメータはモデル番号，拡大率，光源計算，フォグ効果，αモードとなります。今回は時間がありませんので，拡大率以外のパラメータの解説は割愛します。とりあえず，モデル番号，拡大率以外のパラメータは1を入力しておけばよいでしょう。

モデルで自機を動かす

さて，これまで説明してきた命令を使って，前号の講義で行った自機の移動プログラムをモデル・バージョンにしたものがリスト1になります。

前回のテキスト文字を移動したプログラムとのちがいは，つぎのようになっています。

①100 行から120 行

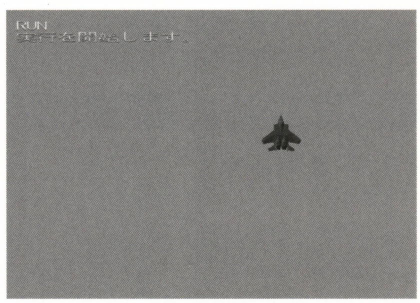
この部分であらかじめ，使用するモデル番号にモデルを割り当てるなどの初期設定を行っています。また，ここですでにモデルを表示状態にしています。

②副プログラム 「P_MOVE」

基本的な処理は変わりませんが，画面外に自機が移動しないように座標の修正を行っている部分が変化しています。使う座標がテキスト座標からワールド座標に変化しているので，画面の端となる座標を変更しています。

《リスト1》モデル・バージョンの自機移動プログラム・リスト

```
100 MODSET 1,7,5                     モデル番号1にモデルバッファ7、オブジェクト番号5のモデルを割り振る
110 MODOPT 1,0.3,1,1,1               モデル番号1のモデルの大きさを変更
120 MODDISP 1                        モデル番号1のモデルを表示状態にする
130 DO
140   CALL P_MOVE                    副プログラム「P_MOVE」を呼び出します。（1000行に移動）
150   CALL WRITE                     副プログラム「WRITE」を呼び出します。（2000行に移動）
160   FSYNC 1
170 LOOP
1000 SUB P_MOVE                      副プログラム「P_MOVE」はここから
1010   IF PAD(1,1)>0 THEN PX=PX＋1   自機の移動処理部分
1020   IF PAD(1,2)>0 THEN PX=PX－1
1030   IF PAD(1,3)>0 THEN PY=PY－1
1040   IF PAD(1,4)>0 THEN PY=PY＋1
1050   IF PX<-56 THEN PX=-56         画面外に移動しないように座標を修正
1060   IF PX>56 THEN PX=56
1070   IF PY<-39 THEN PY=-39
1080   IF PY>39 THEN PY=39
1090 END SUB                         副プログラム「P_MOVE」おわり。（120行に戻る）
2000 SUB WRITE                       副プログラム「WRITE」はここから
2010   MODPOS 1,PX,PY,0              モデル番号1のモデルの位置をPX,PY,0に指定
2020 END SUB                         副プログラム「WRITE」おわり。（130行に戻る）
```



▲モデルを使った自機移動画面（リスト1を実行）。この画面では背景色を変えています

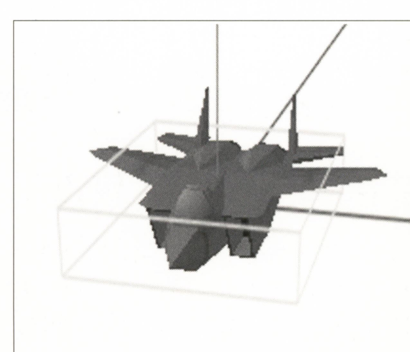
なお，画面の端となるワールド座標はモデルとカメラの位置関係で変わります。画面の端の座標を調べるには，メインループ内にPRINT 文を入れて，PX とPY を表示させるとよいでしょう。

③副プログラム「WRITE」

表示する物が変わったため，大幅に変更を行っています。テキスト表示による自機の移動プログラムでは画面をクリアして古い座標の自機を消去してから新しい座標に自機を表示することで，自機の移動を表現していました。しかし，モデルを使った場合は，現実世界で物を動かすように，新しい座標を指定すれば以前の座標にモデルが残ることはありません。このため，ここではモデルの位置を新しい座標に指定するだけとなっています。

当たり判定の設定

前号のテキスト版シューティングでは弾と敵の座標をチェックして，同じ座標だった場合に「当たっている」と判断していました。今回のモデル版シューティングでも同様の方法を使用してもいいのですが，じつはもっと


《第6図》戦闘機を囲っている枠が当たり判定

《第３表》今回使ったシューティングのモデル番号表（B_MAX=3，E_MAX=5のとき）

番号	モデルの役割	モデルの役割その２
1	自機	自機爆発球
11～13	弾	なし
21～25	敵	敵爆発球

簡単な方法があります。

BASIC STUDIOではモデルを作成する際に基準となる当たり判定をつけることができます。この当たり判定をチェックする命令や関数を使えば，より簡単に当たり判定をチェックすることができるのです（第6図）。

当たり判定のチェックにはCOLLISION関数を使います。これは特定のモデルが，別に指定したモデル（複数設定することができます）と当たっている場合，衝突しているモデルのモデル番号を返してくれる関数です。関数の書きかたはつぎのようになります。

COLLISION（当たり判定をするモデル番号,判定開始モデル番号,判定終了モデル番号）

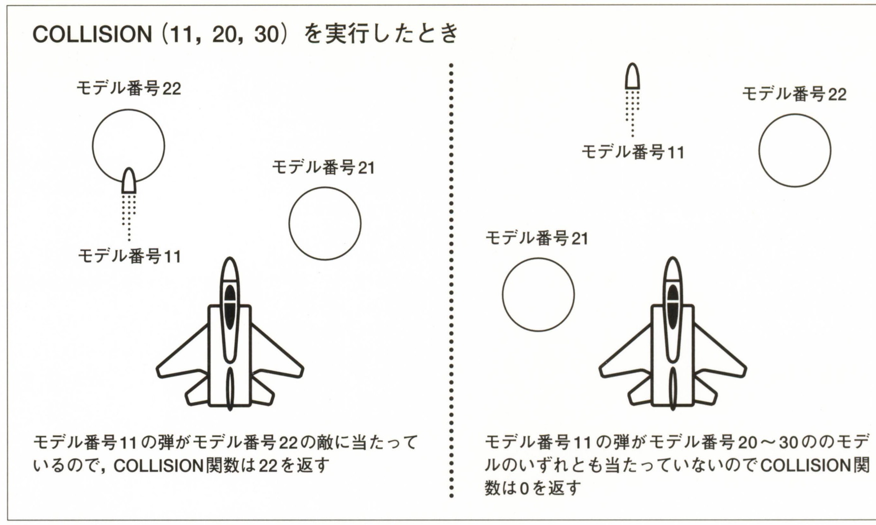
たとえば，当たり判定をするモデル番号のモデルに弾のモデル番号の11を，判定開始モデル番号から判定終了モデル番号までのモデルに敵のモデル番号である20～30を設定したとします。

すると，COLLISION関数で弾に敵が当たっているかどうかがチェックされ，衝突している敵があればその敵のモデルの番号を返してくれます。また，衝突していない場合はモデル番号の替わりに0が返ってきます（第7図）。

シューティング・ゲームをポリゴンモデルを使ったものに変更

さて，これまで説明してきた命令や関数を使うことで，前回作成したシューティングをポリゴンモデルを使ったものに改造することができます。

ポリゴンモデルを使ったプログラムはリスト２のようになります。



《第７図》COLLISION関数による当たり判定チェック

リスト２解説

基本的なちがい

①モデル番号の扱い

モデルを扱うには，扱いたいモデルにモデルの管理番号を割り振る必要があるということは前述したとおりです。ここでは，1を自機のモデル番号，弾の管理番号に10を足した番号を弾のモデル番号，敵の管理番号に20を足した番号を敵のモデル番号としています。よって「弾の管理番号＝弾のモデル番号」（「敵の管理番号＝敵のモデル番号」）とはならないことに注意してください。

また，自機と敵が爆発した際は，爆発用のモデル番号を用意するのではなく，MODSETを使って自機や敵のモデル番号に爆発球を関連付けています（第３表）。

②110～130行

ここでは，モデルの拡大率や画面の広さを変数を使って設定しています。拡大率や画面の広さを変えたい場合はこの設定値を変更するだけで簡単に変更できるようにしています。

③240～420行

テキスト版と同様に弾と敵の初期設定を行っています。モデル版ではさらにモデル番号の割り振り，モデルの向き，モデルの拡大率が加わっています。

④430～500行

自機の初期設定も弾などと同じようにモデルの初期設定などが加わっています。

⑤600～800行　メインループ

メインループ部分はテキスト版とほとんど同じです。とくに説明の必要はないでしょう。

⑥1010～1100行　プレイヤー移動処理

ここもテキスト版と変わりません。縦と横に動くことのできる限界値は120，130行で設定した変数を使って処理しています。

⑦2000～2220行　描画処理

描画処理部分となっていますが，ここでは移動処理部分で変更された座標をMODPOS命令を使ってモデルに反映する部分となっています。また，敵などが爆発したときの爆発球の拡大もここで行っています。

例：2030～2060行

自機の処理部分。自機の状態によって爆発球の拡大率を反映したり（2030行），無敵状

《第４表》今回の講座で使用した命令，関数一覧（登場順）

種類	名前	基本的な書式	役割
命令	MODDISP	MODDISP モデル番号	モデル番号のモデルを表示状態にする
命令	MODOFF	MODOFF モデル番号	モデル番号のモデルを非標示状態にする
命令	MODSET	MODSET モデル番号，バッファ番号，オブジェクト番号	モデル番号に指定したバッファ番号，オブジェクト番号のモデルを割り振る
命令	MODPOS	MODPOS モデル番号，X座標，Y座標，Z座標	モデル番号のモデルの位置を指定した座標に変更する
命令	MODROT	MODROT モデル番号，X回転角，Y回転角，Z回転角	モデル番号のモデルの向きを指定した角度に変更する
関数	RAD	RAD（角度）	括弧内に書かれた数字を度数表記からラジアンに変更する
命令	MODOPT	MODOPT モデル番号，拡大率，光源計算，フォグ効果，αモード	モデル番号のモデルの各種設定を行なう
関数	COLLISION	COLLISION（当たり判定するモデル番号，判定開始モデル番号，判定終了モデル番号）	当たり判定するモデルに判定開始から判定終了までのモデルが当たっていたらモデル番号を返す
命令	CAMPOS	CAMPOS カメラ番号，X座標，Y座標，Z座標	カメラ番号のカメラの位置を指定した座標に変更する
命令	CAMROT	CAMROT カメラ番号，X回転角，Y回転角，Z回転角	カメラ番号のカメラの向きを指定した座標に変更する
命令	BGCOL	BGCOL R成分，G成分，B成分	背景色を変更する

《第5表》配列，変数一覧

	名前	役割
自機関連	PX	自機のX座標
	PY	自機のY座標
	P_FLAG	自機の状態フラグ 1：通常 2：爆発 3：無敵（復活時）
	P_EXPTIME	自機の爆発時間用カウンタ
	MUTEKI_TIME	自機の無敵時間用カウンタ
弾関連	B_MAX	弾の最大数
	B_STORAGE	弾の管理番号格納用配列
	B_X	弾のX座標
	B_Y	弾のY座標
	B_FLAG	弾の状態フラグ 0：弾画面内になし 1：弾画面内にあり
	B_COUNT	B_STORAGEと組み合わせて弾の管理番号の振り分けに使う
	B_ID	弾の管理番号
	B_INTER	弾の発射間隔調整用カウンタ
敵関連	E_MAX	敵の最大数
	E_STORAGE	敵の管理番号格納用配列
	E_X	敵のX座標
	E_Y	敵のY座標
	E_FLAG	敵の状態フラグ 0：敵画面内にいない 1：敵画面内にあり 2：爆発状態
	E_COUNT	E_STORAGEと組み合わせて敵の管理番号の振り分けに使う
	E_ID	敵の管理番号
	E_EXPTIME	敵の爆発時間
	E_INTER	敵の登場間隔調整用カウンタ
その他	SCALE	基準となる拡大率
	SX	左右に移動できる範囲
	SY	上下に移動できる範囲
	MISS	自機がやられた回数
	SCORE	得点

態を表すために一定時間自機を明滅させています（2050行）。

⑧3000～3130行 弾発射処理

基本的にはテキスト版と同じなのですが，弾が発射されたときに弾のモデルをMODDISPで表示状態にしています。

⑨4000～4120行 弾移動処理

これも基本的にはテキスト版と変わりません。弾が画面外に移動した場合にMODOFFでモデルを非表示状態にしているぐらいです。

⑩5000～6120行 敵出現処理，敵移動処理

変更点は弾の発射処理，移動処理と同じく，敵が出現したらモデルを表示状態に，敵が画面外に移動したら非表示状態にしているくらいです。なお，自機や弾と違い，敵のZ座標は0ではなく2となっています（2130，5070行）。これは，使っているモデルの形が正面から見て「T」の形をしているため（当たり判定も同じ形をしています），敵の座標を少しずらして翼にも弾が当たるようにしているのです。

⑪7000～7240行 ヒットチェック

当たり判定の処理部分では，前述のCOLLISION関数を使っています。弾に当たった敵のモデル番号を変数HIT_E_NOに代入し（7020行），弾が敵に当たっていたら，このモデル番号を使って敵の状態フラグを変化させたり（7090行），敵のモデルを爆発球に変えたりしています（7110行）。この処理をFOR～NEXT文を使ってすべての弾で行っています。

⑫8000～8150行 敵爆発処理

敵が弾に当たって爆発した際の処理です。爆発時間を扱う配列E_EXPLOの値を変化させたり，爆発が終わったときには，爆発球のモデルを敵のモデルに戻していたりします（8040～8120行）。

⑬9000～9140行 自機爆発処理

自機と敵が当たって爆発した際の処理です。敵の爆発処理とほぼ同じですが，敵とちがい，自機は一定時間の無敵状態になる処理が追加されています（9060行，9110行）。

⑭10000～10070行 ゲーム・オーバー処理

こちらはテキスト版と変わりません。

⑮11000～11070行 無敵処理

こちらもテキスト版と変わりません。

カメラを設定する

さて，先ほどのモデル版シューティングはモデルを使用していたものの，なんとなく2Dっぽい印象があります。また，背景が暗いので，自機や敵が見えづらくなっています。そこで，カメラと背景色を設定することで，もう少し3Dシューティングらしくしてみましょう。

カメラの設定にはモデルと同じように「位置の指定」と「向きの指定」があります。さきほどのシューティング・ゲームのカメラを**第8図**のように変更してみましょう。なお，カメラは2台あるのですが，通常は1番のカメラを使用しています。

カメラの位置を設定する

カメラの位置を変更するには「CAMPOS」命令を使用します。位置の指定はモデルの位置を指定するのと同様にワールド空間座標で行います。書きかたは，つぎのようになります。

```
CAMPOSカメラ番号， ワールド空間x座標， ワールド空間y座標， ワールド空間z座標
```

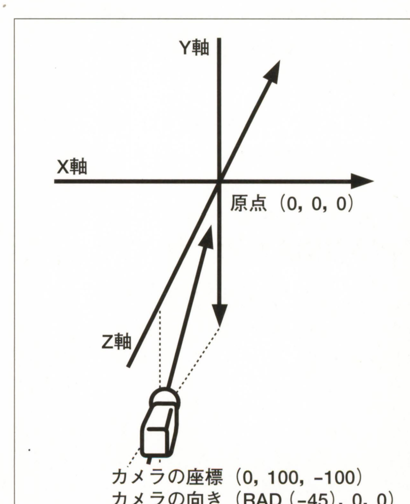
今回はカメラの位置を（0,100,-100）にしたいので，

```
CAMPOS 1，0，100，-100
```

と入力します。

カメラの向き

つぎにカメラの向きを設定します。カメラの向きもモデルと同様にX軸，Y軸，Z軸を軸とした回転角を指定することで設定できます。書きかたは，つぎのようになります。



《第8図》カメラ位置を変更する

```
CAMROT カメラ番号,X軸回転角,Y軸回転角,Z軸回転角
```

先ほどの**第8図**を見るとX軸を軸として－45度上を向かせたいので

```
CAMROT 1,RAD(-45),0,0
```

となります。

背景色を変える

背景色の設定にはBGCOLという命令を使います。これは，色の赤成分，緑成分，青成分を混ぜ合わせて背景色を設定します。書きかたはこのようになります。

```
BGCOL 赤成分,緑成分,青成分
```

今回は空の色に設定したいので，

```
BGCOL 100,130,200
```

とします。
これらの命令を先ほどのプログラムに

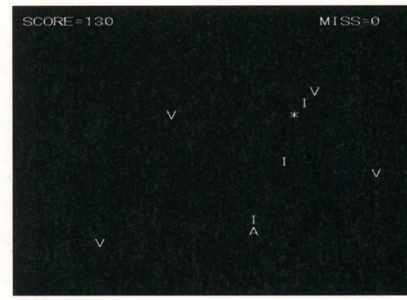
```
510 CAMROT 1,RAD(-45),0,0
520 CAMPOS 1,0,100,-100
530 BGCOL 100,130,200
```

として付け加えれば，3Dシューティング・ゲームは完成です。

改造してみよう！！

さて，今回の講義で作ってみたポリゴンモデルによるシューティング・ゲームはいかがでしょうか？表示方式がモデルになったものの，基本的なアルゴリズムはテキスト文字を使って作成したものと変わらないことがわかると思います。

また，今回作成したシューティング・ゲームは基本的な要素しか搭載していないため，ゲームとしてはつまらないものになっていると思います。しかし，プログラミングの良いところは，元となるゲーム・プログラムの改造ができることです。作成したシューティン



▲前号のテキスト・バージョンのシューティング・ゲーム

《リスト2》ポリゴンモデルを使ったシューティング・ゲームのプログラム・リスト

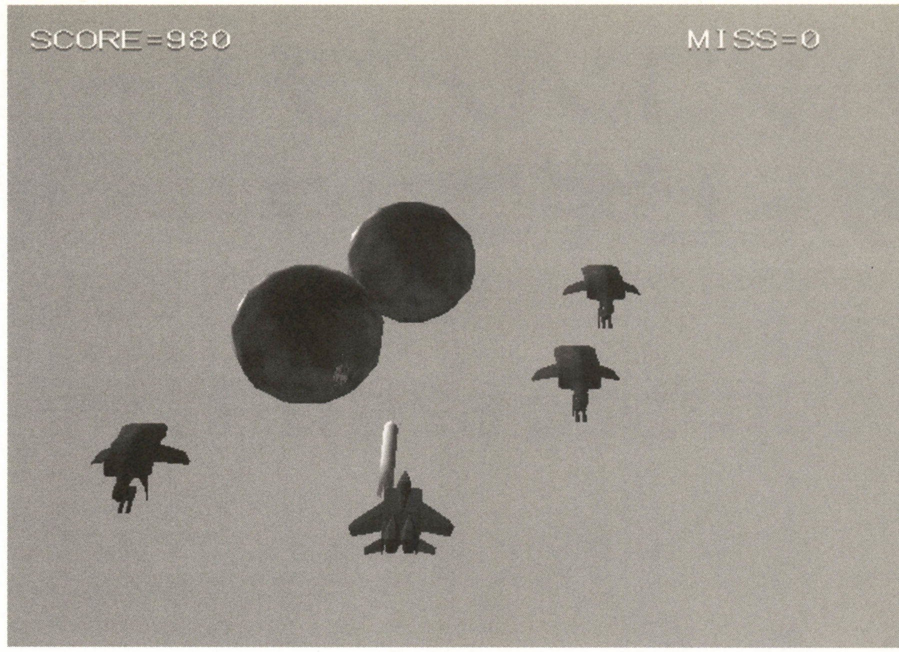
```
100 !初期設定
110 SCALE=0.2
120 SX=58
130 SY=45
140 TEXTMODE 1
150 B_MAX=3! 一画面内の自機の弾数
160 E_MAX=5! 一画面内の敵の数
170 DIM B_STORAGE(B_MAX),B_X(B_MAX),B_Y(B_MAX),B_FLAG(B_MAX)
180 !弾ID格納用,弾のX座標,弾のY座標,弾状態用配列
190 DIM E_STORAGE(E_MAX),E_X(E_MAX),E_Y(E_MAX),E_FLAG(E_MAX)
200 !敵ID格納用,敵のX座標,敵のY座標,敵状態用配*
210 DIM E_EXPTIME(E_MAX)!敵の爆発時間用カウンタ
220 !初期値セット
230 CLEAR
240 B_COUNT=1!画面上の弾数用カウンタ
250 FOR B_ID=1 TO B_MAX!弾ID初期値格納
260  B_FLAG(B_ID)=0
270  B_STORAGE(B_ID)=B_ID
280  MODOFF 10+B_ID
290  MODSET 10+B_ID,5,1
300  MODROT 10+B_ID,-PI/2,PI,0
310  MODOPT 10+B_ID,SCALE,1,1,1
320 NEXT B_ID
330 E_COUNT=1!画面上の敵数用カウンタ
340 FOR E_ID=1 TO E_MAX!敵ID初期値格納
350  E_FLAG(E_ID)=0
360  E_STORAGE(E_ID)=E_ID
370  MODOFF 20+E_ID
380  MODSET 20+E_ID,7,9
390  MODPOS 20+E_ID,-SY-10,-SX-10,0
400  MODROT 20+E_ID,PI/2,0,0
410  MODOPT 20+E_ID,SCALE,1,1,1
420 NEXT E_ID
430 P_FLAG=1
440 PX=0
450 PY=SY-2
460 MODPOS 1,PX,PY,0
470 MODSET 1,7,5
480 MODROT 1,-PI/2,PI,0
490 MODOPT 1,SCALE,1,1,1
500 MODDISP 1
600 !メインループ
610 DO
620  IF P_FLAG=1 OR P_FLAG=3 THEN
630   CALL PLAYER_CTRL
640   CALL PLAYER_SHOT
650  END IF
660  CALL B_MOVE
670  CALL E_ENTRY
680  CALL E_MOVE
690  CALL HIT_CHK
700  IF MISS>=3 THEN CALL GAME_OVER
710  CALL E_EXPRO
720  CALL P_EXPRO
730  CALL MUTEKI
740  B_INTER=B_INTER-1!弾インターバル処理
750  IF B_INTER<0 THEN B_INTER=0
760  E_INTER=E_INTER-1!敵インターバル処理
770  IF E_INTER<0 THEN E_INTER=0
780  FSYNC 1!ウェイト
790  CALL WRITE
800 LOOP
1000 !サブルーチン
1010 SUB PLAYER_CTRL! プレイヤー制御
1020  IF PAD(1,1)>0 THEN PX=PX+1
1030  IF PAD(1,2)>0 THEN PX=PX-1
1040  IF PAD(1,3)>0 THEN PY=PY-1
1050  IF PAD(1,4)>0 THEN PY=PY+1
1060  IF PX<=-SX THEN PX=-SX
1070  IF PX>=SX THEN PX=SX
1080  IF PY<=-SY THEN PY=-SY
1090  IF PY>=SY THEN PY=SY
1100 END SUB
2000 SUB WRITE! 描画処理
2010  CLEAR
2020  MODPOS 1,PX,PY,0
2030  IF P_FLAG=2 THEN MODOPT 1,(90-P_EXPTIME)/45,1,1,1
2040  IF P_FLAG=3 THEN
2050   IF MOD(MUTEKI_TIME,10)<5 THEN MODDISP 1 ELSE MODOFF 1
2060  END IF
2070  FOR B_ID=1 TO B_MAX
2080   IF B_FLAG(B_ID)=1 THEN
2090    MODPOS 10+B_ID,B_X(B_ID),B_Y(B_ID),0
2100   END IF
2110  NEXT B_ID
2120  FOR E_ID=1 TO E_MAX
2130   MODPOS 20+E_ID,E_X(E_ID),E_Y(E_ID),2
2140   IF E_FLAG(E_ID)=2 THEN
2150    MODOPT 20+E_ID,SCALE+(30-E_EXPTIME(E_ID))/45,1,1,1
2160   END IF
2170  NEXT E_ID
2180  LOCATE 29,0
2190  PRINT "MISS=";MISS
2200  LOCATE 0,0
2210  PRINT "SCORE=";SCORE
2220 END SUB
3000 SUB PLAYER_SHOT! 弾発射処理
3010  IF PAD(1,6)>0 AND B_INTER=0 THEN
3020   IF B_COUNT>B_MAX THEN EXIT SUB
3030   B_ID=B_STORAGE(B_COUNT)
3040   B_FLAG(B_ID)=1
3050   B_X(B_ID)=PX
3060   B_Y(B_ID)=PY
3070   MODPOS 10+B_ID,B_X(B_ID),B_Y(B_ID),0
3080   MODDISP 10+B_ID
3090   B_INTER=10
3100   B_COUNT=B_COUNT+1
3110   SEPLAY 18,200,200
3120  END IF
3130 END SUB
4000 SUB B_MOVE! 弾移動処理
4010  FOR B_ID=1 TO B_MAX
```

グ·ゲームの不満点やつまらないところをあなたの手で一つ一つ改良してみてください。こうすることで，プログラムに対する理解が一層深まるはずです。

最後に

さて，これまで3回にわたって連載してきましたが，ゲーム·プログラミングについてなんとなく理解することができましたでしょうか？

前回も説明しましたが，プログラミングを行う際には「作りたいゲームの要素を洗い出す」「洗い出した要素をBASICに翻訳する」といった二つの手順を踏めばそうむずかしいことではないと思います。今回の連載で学んだことをもとに，つぎはあなただけの独創的なゲームのプログラミングに挑戦してみてください。



▲今月のポリゴンモデルを使用したシューティング・ゲーム（リスト2実行画面。ただしモデルの大きさを2倍にしています）

```
4020   IF B_FLAG(B_ID)=1 THEN
4030    B_Y(B_ID)=B_Y(B_ID)-2.5
4040    IF B_Y(B_ID)<-50 THEN
4050     B_FLAG(B_ID)=0
4060     MODOFF 10+B_ID
4070     B_COUNT=B_COUNT-1
4080     B_STORAGE(B_COUNT)=B_ID
4090    END IF
4100   END IF
4110  NEXT B_ID
4120 END SUB
5000 SUB E_ENTRY! 敵出現処理
5010  IF E_INTER=0 THEN
5020   IF E_COUNT>E_MAX THEN EXIT SUB
5030   E_ID=E_STORAGE(E_COUNT)
5040   E_FLAG(E_ID)=1
5050   E_X(E_ID)=INT(SX-2*RND*SX)
5060   E_Y(E_ID)=-SY
5070   MODPOS 20+E_ID,E_X(E_ID),E_Y(E_ID),2
5080   MODDISP 20+E_ID
5090   E_INTER=10
5100   E_COUNT=E_COUNT+1
5110  END IF
5120 END SUB
6000 SUB E_MOVE! 敵移動処理
6010  FOR E_ID=1 TO E_MAX
6020   IF E_FLAG(E_ID)=1 THEN
6030    E_Y(E_ID)=E_Y(E_ID)+0.3
6040    IF E_Y(E_ID)>SY THEN
6050     E_FLAG(E_ID)=0
6060     MODOFF 20+E_ID
6070     E_COUNT=E_COUNT-1
6080     E_STORAGE(E_COUNT)=E_ID
6090    END IF
6100   END IF
6110  NEXT E_ID
6120 END SUB
7000 SUB HIT_CHK! ヒットチェック
7010  FOR B_ID=1 TO B_MAX
7020   HIT_E_NO=COLLISION(10+B_ID,20,20+E_MAX)
7030    IF HIT_E_NO<>0 AND E_FLAG(HIT_E_NO-20)=1 AND B_FLAG(B_ID)=1 THEN
7040     SCORE=SCORE+10
7050     B_FLAG(B_ID)=0
7060     MODOFF 10+B_ID
7070     B_COUNT=B_COUNT-1
7080     B_STORAGE(B_COUNT)=B_ID
7090     E_FLAG(HIT_E_NO-20)=2
7100     MODOFF HIT_E_NO
7110     MODSET HIT_E_NO,5,9
7120     MODDISP HIT_E_NO
7130     E_EXPTIME(HIT_E_NO-20)=30
7140     SEPLAY 23,200,200
7150    END IF
7160  NEXT B_ID
7170   PHIT_E_NO=COLLISION(1,20,20+E_MAX)
7180   IF PHIT_E_NO<>0 AND P_FLAG=1 AND E_FLAG(PHIT_E_NO-20)=1 THEN
7190     P_FLAG=2
7200     MODSET 1,5,9
7210     P_EXPTIME=60
7220     SEPLAY 23,200,200
7230    END IF
7240  END SUB
8000 SUB E_EXPRO!敵爆発処理
8010  FOR E_ID=1 TO E_MAX
8020   IF E_FLAG(E_ID)=2 THEN
8030    E_EXPTIME(E_ID)=E_EXPTIME(E_ID)-1
8040    IF E_EXPTIME(E_ID)<=0 THEN
8050     E_EXPTIME(E_ID)=0
8060     E_FLAG(E_ID)=0
8070     MODOFF 20+E_ID
8080     MODSET 20+E_ID,7,9
8090     MODOPT 20+E_ID,SCALE,1,1,1
8100     E_COUNT=E_COUNT-1
8110     E_STORAGE(E_COUNT)=E_ID
8120    END IF
8130   END IF
8140  NEXT E_ID
8150 END SUB
9000 SUB P_EXPRO!自機爆発処理
9010  IF P_FLAG=2 THEN
9020   P_EXPTIME=P_EXPTIME-1
9030   IF P_EXPTIME<=0 THEN
9040    P_EXPTIME=0
9050     MISS=MISS+1
9060     P_FLAG=3
9070     MODOFF 1
9080     MODSET 1,7,5
9090     MODOPT 1,SCALE,1,1,1
9100     MODDISP 1
9110     MUTEKI_TIME=90
9120    END IF
9130  END IF
9140 END SUB
10000 SUB GAME_OVER! ゲームオーバー処理
10010 CALL WRITE
10020  LOCATE 20,11
10030  PRINT "GAME OVER"
10040  FSYNC 180
10050 OVER_CNT=OVER_CNT+1
10060  END
10070 END SUB
11000 SUB MUTEKI! 無敵処理
11010  IF P_FLAG=3 THEN
11020   MUTEKI_TIME=MUTEKI_TIME-1
11030   IF MUTEKI_TIME<=0 THEN
11040    P_FLAG=1
11050   END IF
11060  END IF
11070 END SUB
```

パソコンレクチャー ザ・ビギニング

Step2：「な2はなくとも2進数!?」

by くりひろし





10進数⟷2進数　対応表

10進	2進	10進	2進	10進	2進	10進	2進	10進	2進
0	0	8	1000	16	10000	24	11000	32	100000
1	1	9	1001	17	10001	25	11001	50	110010
2	10	10	1010	18	10010	26	11010	100	1100100
3	11	11	1011	19	10011	27	11011	200	11001000
4	100	12	1100	20	10100	28	11100	300	100101100
5	101	13	1101	21	10101	29	11101	400	110010000
6	110	14	1110	22	10110	30	11110	500	111110100
7	111	15	1111	23	10111	31	11111	600	1001011000

2進法の基本

・使う数字は0と1のみ

・計算は下のようになる

0+0=0

0+1=1

1+1=10

(2がないのですぐくりあがり)

0−0=0

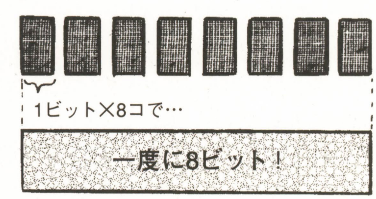
1−0=1

1−1=0

10−1=1

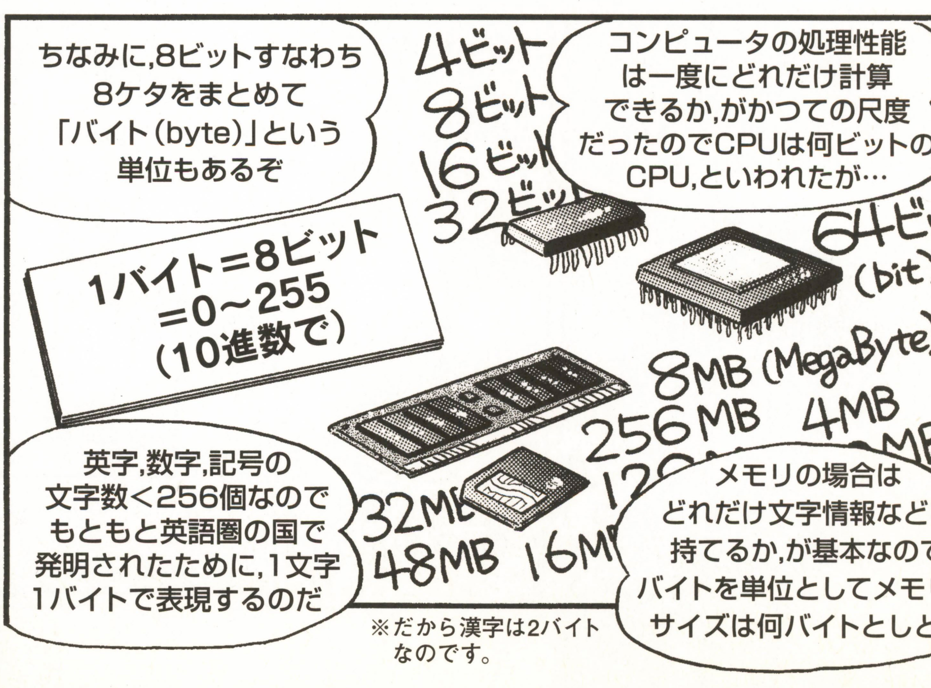


そして,このうち1ケタの0か1だけの2進数で表現できる情報の最小単位を「ビット(bit)」という

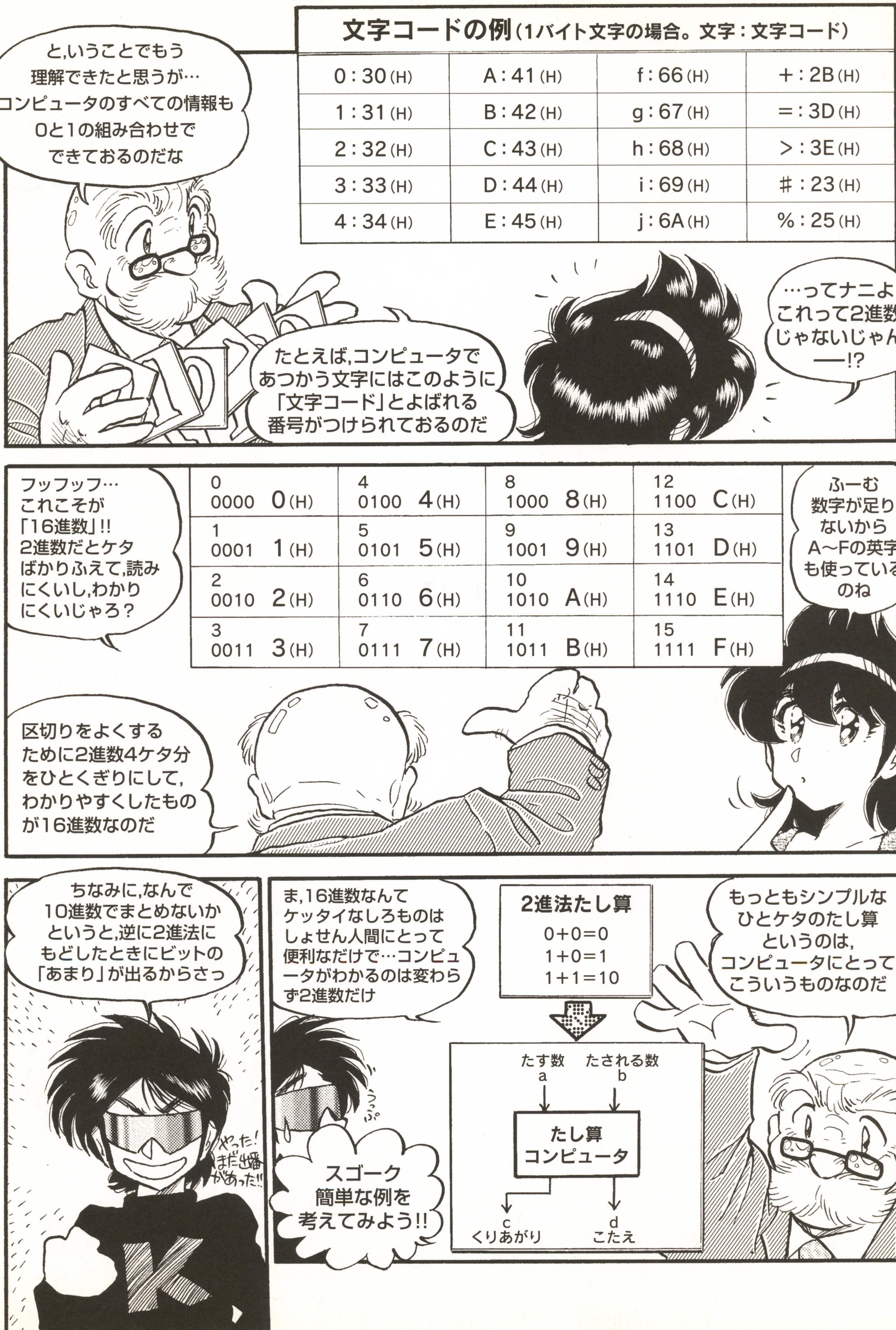


したがって,たとえば2進数で一度に8ケタまで――すなわち8ビットの数をあつかえるコンピュータを8ビットコンピュータとか8ビットCPUとか呼んでいたのだな





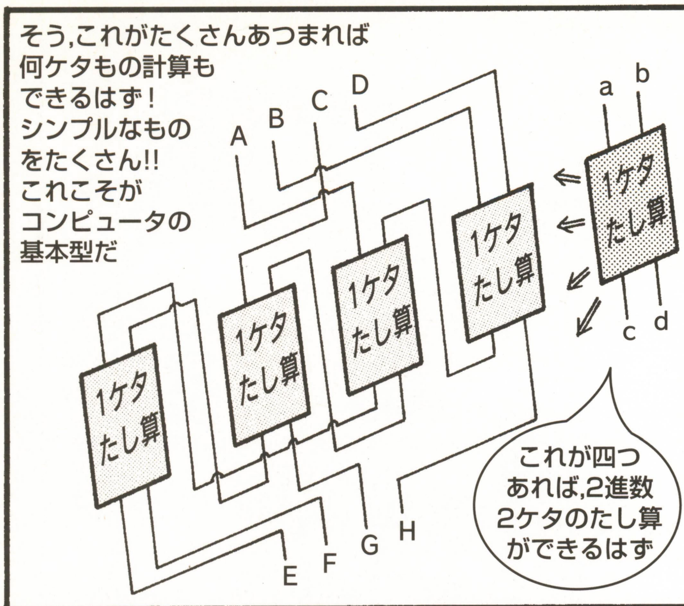
※だから漢字は2バイトなのです。



文字コードの例（1バイト文字の場合。文字：文字コード）

0：30(H)	A：41(H)	f：66(H)	＋：2B(H)
1：31(H)	B：42(H)	g：67(H)	＝：3D(H)
2：32(H)	C：43(H)	h：68(H)	＞：3E(H)
3：33(H)	D：44(H)	i：69(H)	＃：23(H)
4：34(H)	E：45(H)	j：6A(H)	%：25(H)

0 0000 0(H)	4 0100 4(H)	8 1000 8(H)	12 1100 C(H)
1 0001 1(H)	5 0101 5(H)	9 1001 9(H)	13 1101 D(H)
2 0010 2(H)	6 0110 6(H)	10 1010 A(H)	14 1110 E(H)
3 0011 3(H)	7 0111 7(H)	11 1011 B(H)	15 1111 F(H)



2進 AB	2進 CD	2進 EFGH	10進 (AB)	10進 (CD)	10進 (EFGH)
00	00	0000	0	0	0
00	01	0001	0	1	1
00	10	0010	0	2	2
00	11	0011	0	3	3
01	00	0001	1	0	1
01	01	0010	1	1	2
01	10	0011	1	2	3
01	11	0100	1	3	4
10	00	0010	2	0	2
10	01	0011	2	1	3
10	10	0100	2	2	4
10	11	0101	2	3	5
11	00	0011	3	0	3
11	01	0100	3	1	4
11	10	0101	3	2	5
11	11	0110	3	3	6

※じっさいには出力は3ケタあれば足りますね。





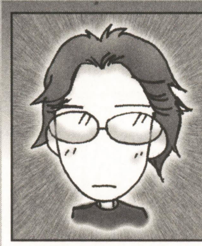
Technical News Spot

マイクロソフトVSボーランドのプログラム開発環境，今年はどちらが勝つ？

テクスポ　2001年10月号

ボーランドは主力開発ツール群に，無償で利用できるバージョンを追加してきました。一方マイクロソフトは開発ツールに，教育向けに特化したバージョンを用意してくるようです。今回は，この2社の動きと，オープン・ソース開発環境の現状をレポートしてみることにしましょう。

TEXT●石川直人



Delphi使い。夏休みは車で，京都ー高松ー福岡ー神戸とまわってきました。一応ノート・パソコンは持っていったのですが，ほとんど触らずに1週間を過ごしました。禁断症状はでませんでした。今月の体重は76.0kg（ー2.0kg）。

レポート

無償で利用できる開発ツールが増加中！

最近のWebアプリケーションで多く使われている技術は，Perl／PHP／Javaのように，無償で利用することができるソフトウエアが基盤となっています。また，プログラミング・ツールではありませんが，Apache，Sendmailのようなサーバ技術や，HTML，DHTML（JavaScript），XMLなどのデータ・フォーマット技術についても，誰でも自由に使うことができるものが多く使われています（これらはオープン・ソースと言われている）。

もともと学術研究目的で開発されてきたWeb（インターネット）の世界では，無償ソフトが広く使われてきました。国や大学などの公的機関に所属する人たちの手で開発されてきたためです。

現在のインターネットの基礎を構成しているのは，そのほとんどが無償ソフトなのです。

マイクロソフトの教育向けVisual Studio .NET

マイクロソフトは，パソコン用のOS（Windows）を核とする世界から，Web（インターネット）を中心にすえた世界へと方向を変えつつあります。

そのための基幹技術が，この連載でも何度も紹介してきた“マイクロソフト・ドットネット”です。

そしてマイクロソフトは，次世代プラットフォームであるドットネットの対応開発ツール『Visual Studio .NET』に，学生向けバージョン『Visual Studio .NET Academic』を用意することを発表しました。

これは，従来のように学生／教員向けに特別な低価格で提供するだけではなく，実際に教員と生徒がプログラミング開発に関する課題をやりとりするための仕組みを用意するなど，パッケージの内容自体が教育向けに特化しているのが特徴です。

Windowsアプリケーション開発の世界では，マイクロソフトの開発ツール群『Microsoft Visual C++』や『Microsoft Visual Basic』が，非常に大きなシェアを占めていました。

しかし，新しいドットネット＝Webアプリケーションの世界では，マイクロソフトの将来性は未知数です。現時点でシェアの多くを占めているオープン・ソース陣営の無償ソフトに取って代わることができるかは，まだわかりません。

ドットネットが次代の標準環境になるためには，まずはドットネット環境用アプリケーションの開発者を増やすことが必要でしょう。新しいネットワーク環境におけるプログラミングに習熟した技術者は，まだまだ少ないのです。

開発ツールの教育分野に対するサポートを強化する戦略は，マイクロソフトのドットネット標準化計画の第1歩といえるでしょう。

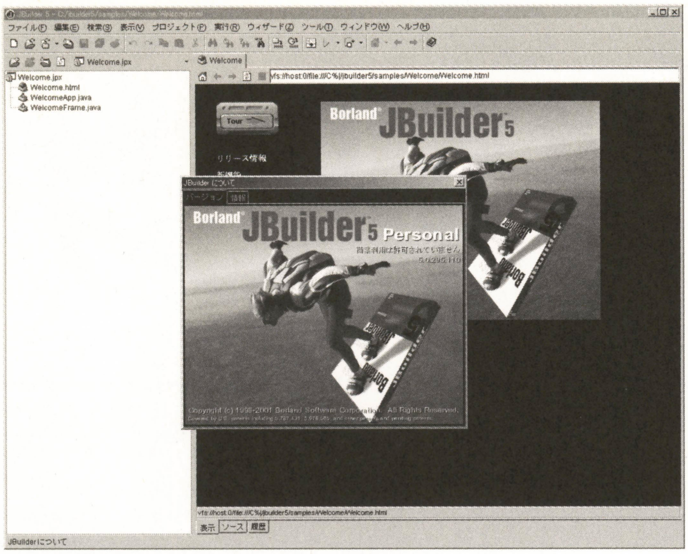
ボーランドのJBuilder／Delphi／Kylix

ボーランドのJava開発ツール『JBuilder』には，かねてからFoundation版という無償で利用できるバージョンが用意されていました。最新のJBuilder 5ではPersonal版と名前が変わりましたが，非商用目的での無償利用が可能な点には変わりありません。

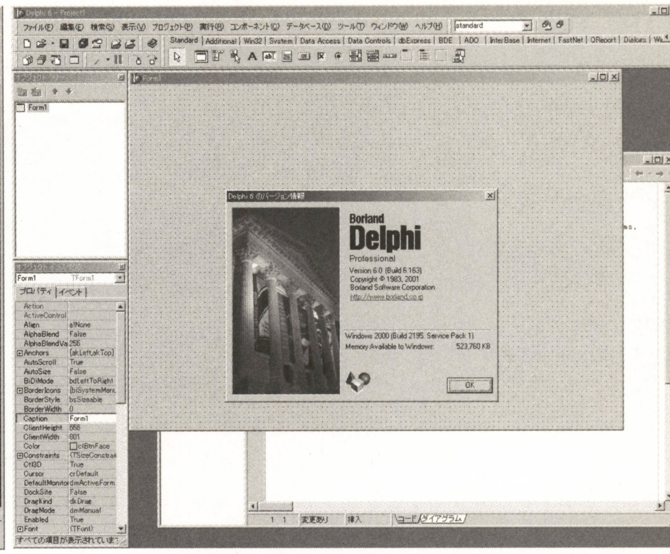
また，ボーランドの新しい開発ツールであるLinux版Delphi『Kylix』にも，Personal版という無償バージョンが発表されています。こちらはすでに英語版の無償配布がはじまっており，以下のアドレスからダウンロードすることができます。

●http://www.borland.com/downloads/

そしてさらに，人気のあるWindows用開発ツール『Delphi』の最新バージョン6においても，ボーランドは無償バージョンを用意することを発表しました。従来は有償で提供されていた最下位バージョ



《写真1》無償で利用できるJava開発ツール『JBuilder 5 Personal』



《写真2》Delphi 6にも無償バージョンが登場する（画面はDelphi 6 Professional）

ンが，今後は非商用目的における無償利用可能になったのです。

以前からボーランドは，開発者の底辺を引き上げるための活動を積極的に行ってきました。ビジュアル開発環境を持たない開発ツール『Borland（ボーランド） C++（シープラスプラス）』も，すでに無償利用が可能になっています。

そして今回ついに，ボーランドの主力開発ツールほとんどにおいて，非商用目的において無償で利用できるバージョンが用意されることになったわけです。

OS開発元という強みがあるマイクロソフトに対して，サード・パーティながらも高い品質の開発ツールで対抗してきたボーランド。その開発ツールのほとんどが無償利用できるようになったことで，利用者は今後ますます増えるのではないでしょうか。

テクスポ 三者三様の立場

インターネットが一般開放され利用者が増えてきたことにより，最近ではインターネットの商用利用が進んできています。その過程で，もともと無償ソフトを基盤に発展してきたインターネットの世界でも，有償ソフトのシェアが増えつつあります。

とくにマイクロソフトは，ドットネット環境の本格的な立ち上げによって，インターネット技術の基盤（サーバ／開発環境）の世界にも一気にシェアを広げようとしています。

また，それ以外にも従来使われてきた無償ソフトと連携して，新しい技術を提供する有償ソフト群も増えてきつつあります。

地道に地歩を固めてきたオープン・ソース陣営による無償ソフト群に対して，先進的な技術を盛り込んだ有償ソフト群（ドットネット）をぶつけてきたマイクロソフト。その他にも，サンやオラクル，IBMなどといった有力各社も，同市場にさまざまな立場でアプローチしており，今後の標準争いは激化することでしょう。

その中でもボーランドは，どちらかというとマイクロソフトとオープン・ソース陣営の中間的な立場を示してきました。

基本的な開発ツールに関しては無償で提供することによって，オープン・ソース陣営に非常に近い立場で，個々人の能力の無償提供による開発を推進します。一方，商用性の強いアプリケーション開発に向いた高級開発ツールに関しては，それなりの価格をつけて有償ツールとして販売するわけです。

今月の予想

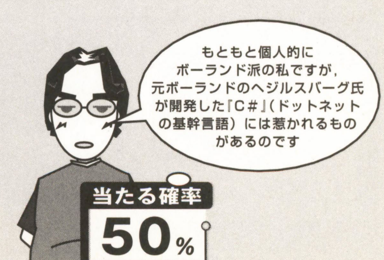
学校の授業にドットネット教育が登場してマイクロソフトが勝つ！

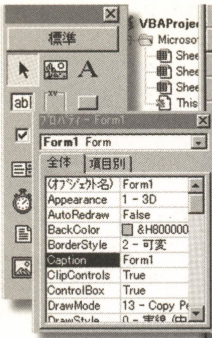
すでにインターネットの世界では実績が十分にあるオープン・ソース陣営は，やはり現時点でもっとも将来性のある存在でしょう。また，すでに有償ソフトとして実績がある開発ツール群を無償化したボーランドも，既存ユーザーからのバックアップが期待でき，ある程度の成功が予想できます。

一方，まったく新しい技術を立ち上げるマイクロソフトですが，まだ詳細がはっきりわからない教育向けバージョンというものの中身が気になります。

いくらマイクロソフトとはいえ，サーバ周りのまったく新しい技術のシェアを一気に広げることは，不可能だと思います。しかし，教育機関からじっくりと新規開発者を育てていくことによって，ドットネットをデファクト・スタンダード化することも不可能ではないでしょう。

大学や専門学校などの授業課目に，Visual Studio .NET Academicを使った課題が氾濫するようになるころには，マイクロソフトがインターネット基幹技術のシェアを制覇しているのです。





はじめよう Microsoft ExcelでのBASICプログラミング

協力 富士通南九州システムエンジニアリング

第6回 toto予想プログラムにチャレンジ（2）

この連載では，Microsoft Excelを使ってBASICのプログラミングを行います。今月は，先月に引き続き簡単なシミュレーション・プログラムを作成します。

TEXT & PROGRAMING 椛島たけし



欧州移籍した選手の活躍を見るために，CSに加入したい。でも我が家の財務省ときたら「毎日遅いのにいつ見るの?」「まだ贅沢品」などと承認してくれないのである。ああ無常。

toto予想プログラム作成

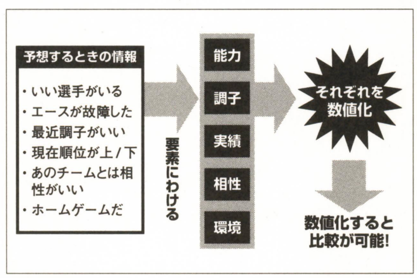
先月より，totoの予想プログラムの作成を行っています。先月は，結果予想の方法とワークシートのデータをコントロールに表示する方法について説明しました。復習する意味で簡単にまとめてみましょう。

- **結果を左右する要素を数値化できれば，シミュレーションが可能（第1図）。…試合結果(得失点差)，順位などを独自の計算式で加工し，結果予想を行います。独自の計算式こそがシミュレーション（結果予想）の心臓部です。**
- **セルのデータはExcelの持つオブジェクト，コレクション，プロパティを利用することで取得可能（第2図）。**

toto予想プログラム

先月までに紹介した方法で，予想の方法やプログラムの作成方法は理解できたと思います。あとはゴリゴリ（笑）プログラミングしていくだけですね。さっそくプログラミングといきたいところですが，ちょっと下準備です。

まずは，Excelのシートに**第3図**のようにデータを入力しておきます。これが予想の元となるデータです。インターネットや新聞などで情報を確認し入力しておきましょう。つぎは，画面デザインです。**第4図**のようにコントロールを配置してください。各コントロールのプロパティは**第1表**のように設定してください。



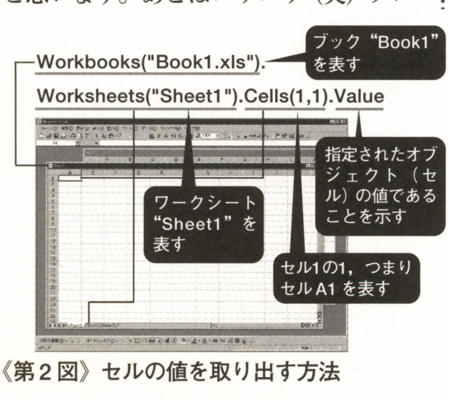
《第1図》シミュレーションには結果を左右する要素の数値化が重要となる

まずは定数を宣言しておこう

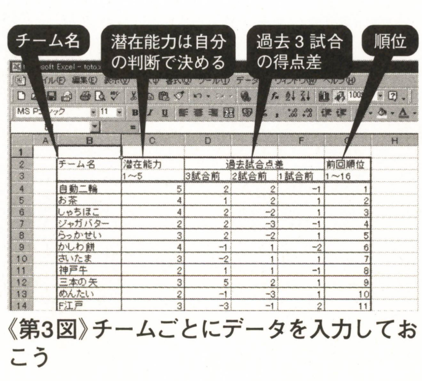
チームの数やチームが登録されているワークシート位置などを，あらかじめ定数として宣言しておきましょう。プログラム内に直接数値を指定するより，メンテナンスがはるかに楽になります。その例が**リスト1**の①部分です。参考にしてください。

チーム名を登録する

まずは，チーム名を表示するコンボボックスにチーム名を登録します。プログラムが起動したときに登録するのがいいですね。したがって，UserFormのInitializeイベントでシートからチーム名を取得して，コンボボックスのAdditemメソッドで追加しましょう。また，予想の要素となる「潜在能力」「調子」「実績」「ホームアドバンテージ」について，どの要素に注目するのか選択できるように，それぞれのコンボボックスに「普通」「重視」

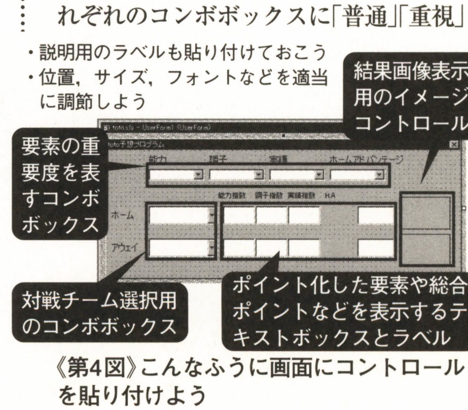


《第2図》セルの値を取り出す方法

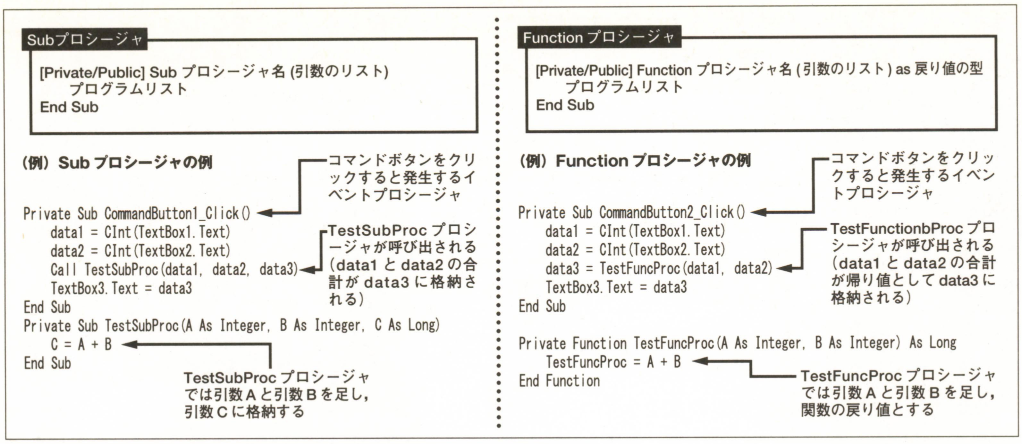


チーム名	潜在能力	過去試合点差			前回順位
	1～5	3試合前	2試合前	1試合前	1～16
自動二輪	5	2	2	-1	1
お茶	4	1	2	1	2
しゃちほこ	4	2	-2	1	3
ジャガバター	2	2	-3	-1	4
らっかせい	3	2	-2	1	5
かしわ餅	4	-1	1	-2	6
さいたま	3	-2	1	1	7
神戸牛	2	1	1	-1	8
三本の矢	3	5	2	1	9
めんたい	2	-1	-3	1	10
F江戸	3	-3	-1	2	11

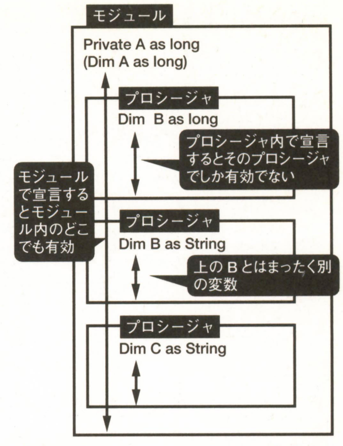
《第3図》チームごとにデータを入力しておこう



《第4図》こんなふうに画面にコントロールを貼り付けよう



《第5図》Sub プロシージャと Function プロシージャの使いかた



《第6図》モジュール内での変数の有効範囲

「軽視」といった項目を追加します。

ワークシートからのデータの読み取りや，コンボボックスへのデータの追加は以前に学習しているのでばっちりですね?! この部分はリスト１の④です。

イベントを活用しよう

コンボボックスでチームを選択すると，チームのポイントが計算されるようにプログラミングを行います。コンボボックスで選択を行うと，Changeイベントが発生します。したがって，ホームチームとアウェイチームそれぞれのコンボボックスのChangeイベントにポイント計算を行うプログラムを行えばいいわけですね。

ではさっそくプログラミングを行ってみましょう。

係数を宣言しておこう

ところで，各要素の計算式は，**第2表**のようにしています。ここで，計算式に使用する「係数」は，定数として宣言しておくことにします。**リスト1の②**のように係数を定数として宣言しておきます。係数がしっくりこないときは，自分なりに変更してください。

プロシージャを作成する

ホームチームとアウェイチームのコンボボックスのChangeイベントでは，つぎの処理を行います。

- **ワークシートからデータを読み込み，チームの基本数値を計算する処理**
- **選択されている要素に応じて基本数値からポイントを計算する処理**
- **ポイントにより画像を表示する処理**

それぞれの処理をプロシージャにわけておくと便利です。プロシージャとは一連の処理をまとめたサブ・ルーチンのようなもので，自由に作成することができます。戻り値を持つものと持たないもので定義方法が多少異なります（第５図）。

では，それぞれの処理をそれぞれプロシージャとして作成します。

ListIndexプロパティを思い出そう

コンボボックスでデータが選ばれたとき，コンボボックスのListIndexプロパティに，選択した項目の番号が格納されます。この番号は，コンボボックスのリ

《第１表》各コントロールの設定

注）設定／変更する必要があるコントロールとプロパティのみ記載してます

各 ComboBox コントロール	
コントロール名	（説明）
cmbTeamHome	ホームチーム
cmbTeamAway	アウェイチーム
cmbPotential	能力
cmbCondition	調子
cmbResults	結果
cmbHome	ホームアドバンテージ
プロパティ設定（共通）	
プロパティ	値
Style	2 - fmStyleDropDownList

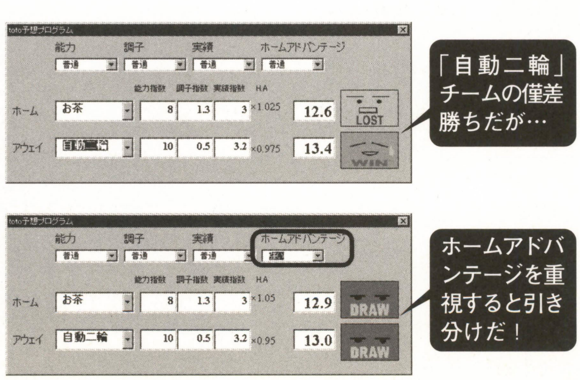
各 TextBox コントロール	
コントロール名	（説明）
txtPotentialHome	能力表示用（ホーム）
txtPotentialAway	能力表示用（アウェイ）
txtConditionHome	調子表示用（ホーム）
txtConditionAway	調子表示用（アウェイ）
txtResultsHome	実績表示用（ホーム）
txtResultsAway	実績表示用（アウェイ）
txtPointHome	合計ポイント表示用（ホーム）
txtPointAway	合計ポイント表示用（アウェイ）
プロパティ設定（共通）	
プロパティ	値
Locked	True

各 Label コントロール	
コントロール名	（説明）
lblHomeHA	ホームアドバンテージ表示用（ホーム）
lblHAwayHA	ホームアドバンテージ表示用（アウェイ）

各 Image コントロール	
コントロール名	（説明）
imgHome	ホームチーム用
imgAway	アウェイチーム用
プロパティ設定（共通）	
プロパティ	値
AutoSize	False
PictureSizeMode	1 - fmPictureSizeModeStretch



《第７図》勝敗が一目でわかる?! 画像ファイルを用意しよう



《第８図》完成プログラムの実行結果

《第２表》各要素の計算式（例）

要素	計算式
能力	潜在能力 × 重要度別能力係数
調子	(3試合前点差 × 3試合前用係数＋2試合前点差 × 2試合前用係数＋1試合前点差 × 1試合前用係数) × 重要度別調子係数
実績	(総チーム数－前回順位) × 重要度別順位係数

ストの順に自動的に割り振られています。したがって，ListIndexプロパティをうまくCellsプロパティに利用すれば，選択したチームの情報や要素の重要度を取得できるはずです。

まずは，各チームの基本数値を計算する処理ですが，要素の重要度を表す「係数」を計算する前の段階まで行います。基本数値など，プロシージャをまたがって使用する変数は，モジュールレベル変数として定義します。変数については**第6図**を参照してください。**リスト1**の⑪⑫部分となります。なお，ホームとアウェイは変数や定数名が変わっただけで実質的には同じプログラムです。

つぎに，要素の重要度に応じて係数を決定し，先ほど計算した基本数値から各要素のポイントを計算します。また，それぞれのポイントを足して，最終的な総合ポイントを算出します。最後に結果画像を表示するプロシージャを呼び出しておきます。**リスト1**の⑬部分です。

残った処理は，画像を表示する部分です。総合ポイントの差に応じて表示する画像を変えることにします（**第7図**のような画像ファイルを用意してみました）。この部分の処理はほとんどが判定処理です。画像をイメージコントロールに表示するにはPictureプロパティとLoad Picture関数を使用します。この方法は8月号で紹介しましたね。**リスト1**の⑭がその部分となります。

イベント処理で呼び出そう

ここまで完成したら，あとはイベント

《リスト1》toto予想のプログラム・リスト

```
Option Explicit
'---------- ①ワークシート用の定数宣言 ----------
Private Const MAX_TEAM_COUNT = 16   'チーム数
Private Const ROW_TEAM_START = 4    'Sheetでチーム名が始まる行
Private Const COL_TEAM = 2          'Sheetでチーム名が存在する列
Private Const COL_POTENTIAL = 3     'Sheetで潜在能力数値が存在する列

Private Const COE_SCORE_MAX = 3     '得失点差の最大値
Private Const COE_SCORE_MIN = -3    '得失点差の最小値

Private Const COL_SCORE_B3 = 4      'Sheet上で3試合前の得失点差が存在する行
Private Const COL_SCORE_B2 = 5      'Sheet上で2試合前の得失点差が存在する行
Private Const COL_SCORE_B1 = 6      'Sheet上で1試合前の得失点差が存在する行

Private Const COL_RESULTS = 7       'Sheet上で前回順位が存在する行

'---------- ②要素の係数を定数宣言 ----------
Private Const COE_POT_HIGH = 3      '能力「重視」の係数
Private Const COE_POT_MID = 2       '能力「普通」の係数
Private Const COE_POT_LOW = 1       '能力「軽視」の係数

Private Const COE_SCORE_B3 = 0.2    '3試合前の得失点差の係数
Private Const COE_SCORE_B2 = 0.3    '2試合前の得失点差の係数
Private Const COE_SCORE_B1 = 0.5    '1試合前の得失点差の係数

Private Const COE_SCORE_HIGH = 1.5  '調子「重視」の係数
Private Const COE_SCORE_MID = 1     '調子「普通」の係数
Private Const COE_SCORE_LOW = 0.5   '調子「軽視」の係数


Private Const COE_RES_HIGH = 0.5    '実績「重視」の係数
Private Const COE_RES_MID = 0.2     '実績「普通」の係数
Private Const COE_RES_LOW = 0.1     '実績「軽視」の係数

Private Const COE_HAD_HIGH = 1.05   'ホームアドバンテージが「強い」係数
Private Const COE_HAD_NR = 1.025    'ホームアドバンテージが「普通」の係数
Private Const COE_HAD_MID = 1       'ホームアドバンテージが「なし」の係数
Private Const COE_HAD_LOW = 0.975   'ホームアドバンテージが「普通」のアウェイ係数
Private Const COE_HAD_HiLOW = 0.95  'ホームアドバンテージが「強い」アウェイ係数

Private Const PATH_BMP = "C:¥ExcelVBA¥toto¥" 'イメージ画像格納先(状況に合わせて変更してください)
Private Const BMP_WIN_WIDE = "winwide.bmp" '大差勝利のイメージファイル名
Private Const BMP_WIN_SLIM = "winslim.bmp" '僅差勝利のイメージファイル名
Private Const BMP_LOST_WIDE = "lostwide.bmp" '大差敗戦のイメージファイル名
Private Const BMP_LOST_SLIM = "lostslim.bmp" '僅差敗戦のイメージファイル名
Private Const BMP_DRAW = "draw.bmp"   '引き分けのイメージファイル名

'---------- ③モジュールレベル変数宣言 ----------
Private dblHomePotentialOrg As Double '潜在能力の基本数値(ホーム)
Private dblHomeConditionOrg As Double '調子の基本数値(ホーム)
Private dblHomeResultsOrg As Double   '実績の基本数値(ホーム)
Private dblAwayPotentialOrg As Double '潜在能力の基本数値(アウェイ)
Private dblAwayConditionOrg As Double '調子の基本数値(アウェイ)
Private dblAwayResultsOrg As Double   '実績の基本数値(アウェイ)

'--- ④フォームロード時の初期処理 ---
Private Sub UserForm_Initialize()

    'ホーム、アウェイチームリストへのチーム名表示
    Dim lngTeamCtr As Long  'チーム数カウント用変数
    For lngTeamCtr = 1 To MAX_TEAM_COUNT
        cmbTeamHome.AddItem Sheet1.Cells(ROW_TEAM_START + (lngTeamCtr - 1), COL_TEAM)
        cmbTeamAway.AddItem Sheet1.Cells(ROW_TEAM_START + (lngTeamCtr - 1), COL_TEAM)
    Next lngTeamCtr

    With cmbPotential    '「潜在能力」リストの表示
        .AddItem "普通"
        .AddItem "重視"
        .AddItem "軽視"
        .ListIndex = 0  'デフォルト表示設定
    End With

    With cmbCondition    '「調子」リストの表示
        .AddItem "普通"
        .AddItem "重視"
        .AddItem "軽視"
        .ListIndex = 0  'デフォルト表示設定
    End With

    With cmbResults      '「実績」リストの表示
        .AddItem "普通"
        .AddItem "重視"
        .AddItem "軽視"
        .ListIndex = 0  'デフォルト表示設定
    End With

    With cmbHome         '「ホームアドバンテージ」リストの表示
        .AddItem "なし"
        .AddItem "普通"
        .AddItem "強い"
        .ListIndex = 0  'デフォルト表示設定
    End With

End Sub

'--- ⑤ホームチーム選択時 ---
Private Sub cmbTeamHome_Change()
    Call procTeamHomeCalc 'ホームチーム基本数値計算処理
    Call procPointCalc    'ポイント計算処理
End Sub

'--- ⑥アウェイチーム選択時 ---
Private Sub cmbTeamAway_Change()
    Call procTeamAwayCalc 'アウェイチーム基本数値計算処理
    Call procPointCalc    'ポイント計算処理
End Sub

'--- ⑦能力の重要度選択時 ---
Private Sub cmbPotential_Change()
    Call procPointCalc   'ポイント計算処理
End Sub

'--- ⑧調子の重要度選択時 ---
Private Sub cmbCondition_Change()
    Call procPointCalc   'ポイント計算処理
End Sub

'---  ⑨実績の重要度選択時 ---
Private Sub cmbResults_Change()
    Call procPointCalc   'ポイント計算処理
End Sub

'--- ⑩ホームアドバンテージの重要度選択時 ---
Private Sub cmbHome_Change()
    Call procPointCalc
End Sub


'--- ⑪ホームチームの基本数値計算 ---
Private Sub procTeamHomeCalc()
    Dim lngTeamHomeRow As Long  'Sheet上の行数値
    Dim lngSCORE_B3 As Long     '3試合前の得失点差
    Dim lngSCORE_B2 As Long     '2試合前の得失点差
    Dim lngSCORE_B1 As Long     '1試合前の得失点差

    '指定チームのWorkSheet上の行数値読み込み
    lngTeamHomeRow = cmbTeamHome.ListIndex + ROW_TEAM_START

    '潜在能力の基本数値読み込み
    dblHomePotentialOrg = Sheet1.Cells(lngTeamHomeRow, COL_POTENTIAL)

    '得失点差が±3を超えた場合、±3とする処理
    lngSCORE_B3 = Sheet1.Cells(lngTeamHomeRow, COL_SCORE_B3)
    If lngSCORE_B3 > COE_SCORE_MAX Then
        lngSCORE_B3 = COE_SCORE_MAX
    ElseIf lngSCORE_B3 < COE_SCORE_MIN Then
        lngSCORE_B3 = COE_SCORE_MIN
    End If

    lngSCORE_B2 = Sheet1.Cells(lngTeamHomeRow, COL_SCORE_B2)
    If lngSCORE_B2 > COE_SCORE_MAX Then
        lngSCORE_B2 = COE_SCORE_MAX
    ElseIf lngSCORE_B2 < COE_SCORE_MIN Then
        lngSCORE_B2 = COE_SCORE_MIN
    End If

    lngSCORE_B1 = Sheet1.Cells(lngTeamHomeRow, COL_SCORE_B1)
    If lngSCORE_B1 > COE_SCORE_MAX Then
        lngSCORE_B1 = COE_SCORE_MAX
    ElseIf lngSCORE_B1 < COE_SCORE_MIN Then
        lngSCORE_B1 = COE_SCORE_MIN
    End If

    '得失点差による調子の基本数値計算
    dblHomeConditionOrg = ((lngSCORE_B3) * COE_SCORE_B3) + _
                    ((lngSCORE_B2) * COE_SCORE_B2) + _
                    ((lngSCORE_B1) * COE_SCORE_B1)

    '前回順位による実績の基本数値計算
    dblHomeResultsOrg = MAX_TEAM_COUNT + 1 - (Sheet1.Cells(lngTeamHomeRow,
COL_RESULTS))

End Sub
```

処理ごとに呼び出すだけです。チーム選択を行った場合に発生する各コンボボックスのChangeイベントで，先ほど作成したプロシージャを記述します（リスト1⑤〜⑧）。また，各要素の重要度が変更されたときには，ポイント計算を行うプロシージャを呼び出すようにします（リスト1⑦〜⑩）。

やっとここまできました。さっそく実行してみましょう。対戦チームごとの予想結果が簡単に表示できます（第8図）。

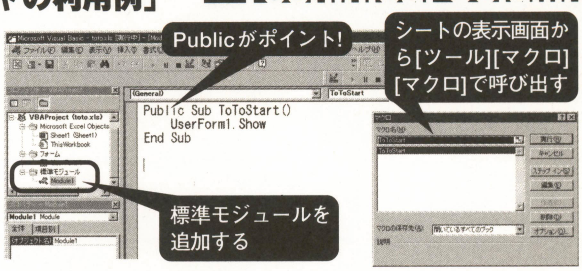
あとは，乱数（Rnd）による番狂わせ処理や下克上処理，など自分なりの要素や係数を作ってみると楽しいでしょう。高確率の予想プログラムができると，シェアウエアとして登録してもいいですね。

さて，来月からは新たなテーマでExcel VBAプログラミングに取り組みたいと思います。おたのしみに。

Techinical Column

「Publicプロシージャの利用例」

標準モジュールにPublicプロシージャを登録すると，ワークシートからマクロとして呼び出すことができる。スタートメニュー表示のマクロなどを登録しておくと便利だ。



▲標準モジュールのプロシージャをマクロとして呼び出す

```
'--- ⑫アウェイチームの基本数値計算 ---
Private Sub procTeamAwayCalc()
    Dim lngTeamAwayRow As Long  'Sheet上の行数値
    Dim lngSCORE_B3 As Long     '3試合前の得失点差
    Dim lngSCORE_B2 As Long     '2試合前の得失点差
    Dim lngSCORE_B1 As Long     '1試合前の得失点差

    '指定チームのWorkSheet上の行数値読み込み
    lngTeamAwayRow = cmbTeamAway.ListIndex + ROW_TEAM_START

    '潜在能力の基本数値読み込み
    dblAwayPotentialOrg = Sheet1.Cells(lngTeamAwayRow, COL_POTENTIAL)

    '得失点差が±3を超えた場合、±3とする処理
    lngSCORE_B3 = Sheet1.Cells(lngTeamAwayRow, COL_SCORE_B3)
    If lngSCORE_B3 > COE_SCORE_MAX Then
        lngSCORE_B3 = COE_SCORE_MAX
    ElseIf lngSCORE_B3 < COE_SCORE_MIN Then
        lngSCORE_B3 = COE_SCORE_MIN
    End If

    lngSCORE_B2 = Sheet1.Cells(lngTeamAwayRow, COL_SCORE_B2)
    If lngSCORE_B2 > COE_SCORE_MAX Then
        lngSCORE_B2 = COE_SCORE_MAX
    ElseIf lngSCORE_B2 < COE_SCORE_MIN Then
        lngSCORE_B2 = COE_SCORE_MIN
    End If

    lngSCORE_B1 = Sheet1.Cells(lngTeamAwayRow, COL_SCORE_B1)
    If lngSCORE_B1 > COE_SCORE_MAX Then
        lngSCORE_B1 = COE_SCORE_MAX
    ElseIf lngSCORE_B1 < COE_SCORE_MIN Then
        lngSCORE_B1 = COE_SCORE_MIN
    End If

    '得失点差による調子の基本数値計算
    dblAwayConditionOrg = ((lngSCORE_B3) * COE_SCORE_B3) + _
                          ((lngSCORE_B2) * COE_SCORE_B2) + _
                          ((lngSCORE_B1) * COE_SCORE_B1)

    '前回順位による実績の基本数値計算
    dblAwayResultsOrg = MAX_TEAM_COUNT + 1 - (Sheet1.Cells(lngTeamAwayRow, COL_RESULTS))

End Sub

'--- ⑬予想ポイント計算 ---
Private Sub procPointCalc()
    Dim dblHomePotential As Double  '変換後の潜在能力(ホーム)
    Dim dblAwayPotential As Double  '変換後の潜在能力(アウェイ)
    Dim dblHomeCondition As Double  '変換後の調子(ホーム)
    Dim dblAwayCondition As Double  '変換後の調子(アウェイ)
    Dim dblHomeResults As Double    '変換後の実績(ホーム)
    Dim dblAwayResults As Double    '変換後の実績(アウェイ)
    Dim dblHomePoint As Double      '変換後のポイント(ホーム)
    Dim dblAwayPoint As Double      '変換後のポイント(アウェイ)
    Dim dblHomePointTmp As Double   'ポイント用作業変数(ホーム)
    Dim dblAwayPointTmp As Double   'ポイント用作業変数(アウェイ)

    '「能力」に関する計算
    Select Case cmbPotential.ListIndex
    Case 0  '「普通」選択
        dblHomePotential = dblHomePotentialOrg * COE_POT_MID
        dblAwayPotential = dblAwayPotentialOrg * COE_POT_MID
    Case 1  '「重視」選択
        dblHomePotential = dblHomePotentialOrg * COE_POT_HIGH
        dblAwayPotential = dblAwayPotentialOrg * COE_POT_HIGH
    Case 2  '「軽視」選択
        dblHomePotential = dblHomePotentialOrg * COE_POT_LOW
        dblAwayPotential = dblAwayPotentialOrg * COE_POT_LOW
    End Select

    '「調子」に関する計算
    Select Case cmbCondition.ListIndex
    Case 0  '「普通」選択
        dblHomeCondition = dblHomeConditionOrg * COE_SCORE_MID
        dblAwayCondition = dblAwayConditionOrg * COE_SCORE_MID
    Case 1  '「重視」選択
        dblHomeCondition = dblHomeConditionOrg * COE_SCORE_HIGH
        dblAwayCondition = dblAwayConditionOrg * COE_SCORE_HIGH
    Case 2  '「軽視」選択
        dblHomeCondition = dblHomeConditionOrg * COE_SCORE_LOW
        dblAwayCondition = dblAwayConditionOrg * COE_SCORE_LOW
    End Select

    '「実績」に関する計算
    Select Case cmbResults.ListIndex
    Case 0  '「普通」選択
        dblHomeResults = dblHomeResultsOrg * COE_RES_MID
        dblAwayResults = dblAwayResultsOrg * COE_RES_MID
    Case 1  '「重視」選択
        dblHomeResults = dblHomeResultsOrg * COE_RES_HIGH
        dblAwayResults = dblAwayResultsOrg * COE_RES_HIGH
    Case 2  '「軽視」選択
        dblHomeResults = dblHomeResultsOrg * COE_RES_LOW
        dblAwayResults = dblAwayResultsOrg * COE_RES_LOW
    End Select

    'ホームチームの数値表示
    txtPotentialHome.Text = dblHomePotential
    txtConditionHome.Text = dblHomeCondition
    txtResultsHome.Text = dblHomeResults
    'アウェイチームの数値表示
    txtPotentialAway.Text = dblAwayPotential
    txtConditionAway.Text = dblAwayCondition
    txtResultsAway.Text = dblAwayResults

    '作業用変数への取り込み
    dblHomePointTmp = dblHomePotential + dblHomeCondition + dblHomeResults
    dblAwayPointTmp = dblAwayPotential + dblAwayCondition + dblAwayResults

    '「ホームアドバンテージ」に関する計算
    Select Case cmbHome.ListIndex
    Case 0  '「なし」選択
        dblHomePoint = dblHomePointTmp * COE_HAD_MID
        dblAwayPoint = dblAwayPointTmp * COE_HAD_MID
        lblHomeHA.Caption = "×1.0"
        lblAwayHA.Caption = "×1.0"
    Case 1  '「普通」選択
        dblHomePoint = dblHomePointTmp * COE_HAD_NR
        dblAwayPoint = dblAwayPointTmp * COE_HAD_LOW
        lblHomeHA.Caption = "×1.025"
        lblAwayHA.Caption = "×0.975"
    Case 2  '「強い」選択
        dblHomePoint = dblHomePointTmp * COE_HAD_HIGH
        dblAwayPoint = dblAwayPointTmp * COE_HAD_HiLOW
        lblHomeHA.Caption = "×1.05"
        lblAwayHA.Caption = "×0.95"
    End Select

    'ホームチーム総合ポイントの表示
    txtPointHome.Text = Format(dblHomePoint, "##,##0.0")
    'アウェイチーム総合ポイントの表示
    txtPointAway.Text = Format(dblAwayPoint, "##,##0.0")

    'イメージ画像の表示
    Call procChangeImage

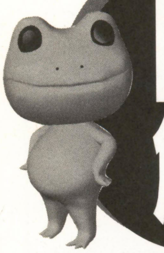
End Sub

'--- ⑭イメージ画像表示プロシージャ ---
Private Sub procChangeImage()
    Dim dblHomePoint As Double      '総合ポイント(ホーム)
    Dim dblAwayPoint As Double      '総合ポイント(アウェイ)
    Dim dblCalcPoint As Double      '判定ポイント

    'ホーム、アウェイの総合ポイントの取り込み
    dblHomePoint = txtPointHome.Text
    dblAwayPoint = txtPointAway.Text

    'ポイント差の判定とビットマップ表示
    Select Case dblHomePoint - dblAwayPoint
    '2点差以上
    Case Is > 2
        imgHome.Picture = LoadPicture(PATH_BMP & BMP_WIN_WIDE)
        imgAway.Picture = LoadPicture(PATH_BMP & BMP_LOST_WIDE)
    '同点より大、2点差未満
    Case 0.5 To 2
        imgHome.Picture = LoadPicture(PATH_BMP & BMP_WIN_SLIM)
        imgAway.Picture = LoadPicture(PATH_BMP & BMP_LOST_SLIM)
    '-2点差以上
    Case Is < -2
        imgHome.Picture = LoadPicture(PATH_BMP & BMP_LOST_WIDE)
        imgAway.Picture = LoadPicture(PATH_BMP & BMP_WIN_WIDE)
    '同点より大、-2点差未満
    Case -2 To -0.5
        imgHome.Picture = LoadPicture(PATH_BMP & BMP_LOST_SLIM)
        imgAway.Picture = LoadPicture(PATH_BMP & BMP_WIN_SLIM)
    '+-0.5点差内
    Case Else
        imgHome.Picture = LoadPicture(PATH_BMP & BMP_DRAW)
        imgAway.Picture = LoadPicture(PATH_BMP & BMP_DRAW)
    End Select

End Sub
```



プログラマーをめざす人の必見講座

図解！ Visual Basic ゲーム作成術

第7回　キャラクター・アクション・ゲーム①フィールド上のキャラクター移動

ディスプレイ画面上に表示されたゲーム・フィールドの中を，キャラクターが動きまわる“キャラクター・アクション・ゲーム”。この種のゲームは，いろいろな仕掛けなどを組み込みやすく，自由度の高いゲーム・デザインができることが特徴です。今月から何回かにわたって，キャラクター・アクション・ゲームを『Microsoft Visual Basic（以下VBと略）』で作成していきましょう。

TEXT&PROGRAMMING●東　幸太



BASIC歴15年以上のプログラマー。今年は猛暑だった。これだけ暑くなると，空調なしの部屋でのパソコン操作は危険になってくる。マニュアルによると，たいていのパソコンの動作が補償される上限温度は35度。それを超えると，熱暴走などが起こりかねない。とはいえ，普通だと人間のほうが先にダウン？

Microsoft Visual Basic Programming System for Windows Ver.6.0（日本語版）

オープン価格
※編集部の価格調査によると，下記価格が目安になります。
ラーニング版2万円強
（アカデミック・パッケージ1万円弱）
プロフェッショナル版8～9万円
（アカデミック・パッケージ1万円強）
問い合わせ先
マイクロソフト
☎03-5454-2300
ホームページ
http://www.microsoft.com/japan/
動作環境
Windows 95／98／2000／NT4.0が動作するマシン



キャラクター・アクション・ゲームの基本構成

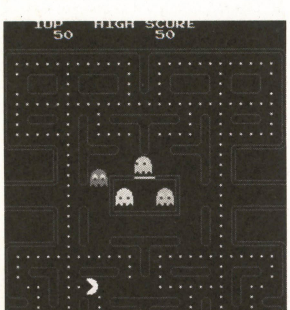
キャラクター・アクションは，大ヒットゲーム『パックマン』（写真1）によって確立され，その後多数のゲームが登場したジャンルです。

特徴として，ゲーム・フィールドに，プレイヤー・キャラクターや敵キャラクター，そして中立的なキャラクターなどが登場します。そして，「敵キャラクターを撃つ」という単純な方法ではなく，アイデアを駆使した方法で攻撃するというものが中心です。敵キャラクターへの攻撃方法が，弾によるものだけになりがちなシューティング・ゲームと異なり，いろいろなルールを設けることができます。

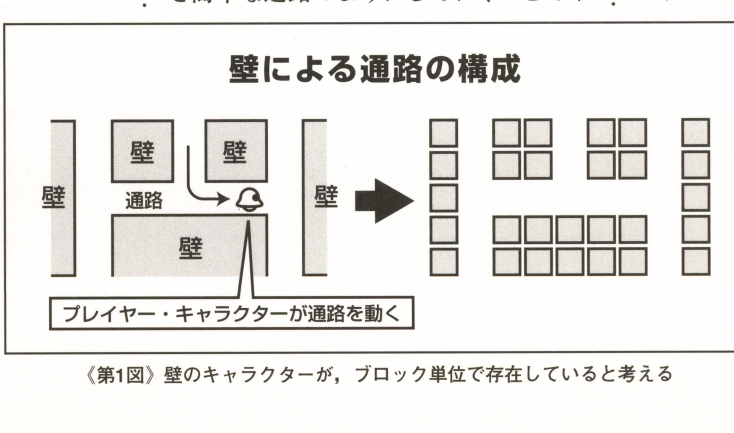
また，ゲームによっては敵キャラクターを攻撃することが主目的ではなく，ゲーム・フィールドに散らばったアイテム・キャラクターを取ったりすることが主目的になっているものもあります。このように，キャラクター・アクション・ゲームはゲーム・デザインの自由度が非常に高いことから，いろいろな要素を取り入れてゲームを構成することが可能です。

各キャラクターは，基本的にブロック単位で構成されたゲーム・フィールド上を移動します。ゲーム・フィールド上に障害物がほとんどないシューティング・ゲームとは異なり，キャラクター・アクション・ゲームでは“壁”などの障害物によって“通路”が構成され，その部分をキャラクターが通っていくようになっています（第1図）。

各キャラクターは，上下左右に移動することができますが，壁などの通れない部分は移動することができません。通路を簡単な迷路のようにしておくことでキャラクターの移動範囲を制限し，それを活用することでゲームをより簡単にクリアできるようにする，といったゲーム・デザインができます。



《写真1》ナムコ『パックマン』。続編も数多く制作されており，最近では携帯電話などでも遊べるようになっている

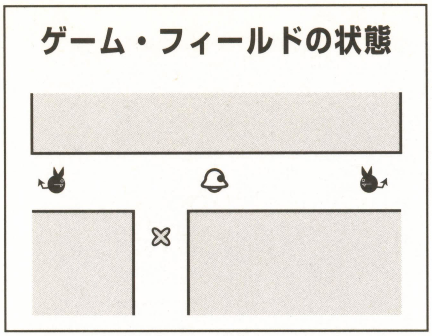


《第1図》壁のキャラクターが，ブロック単位で存在していると考える

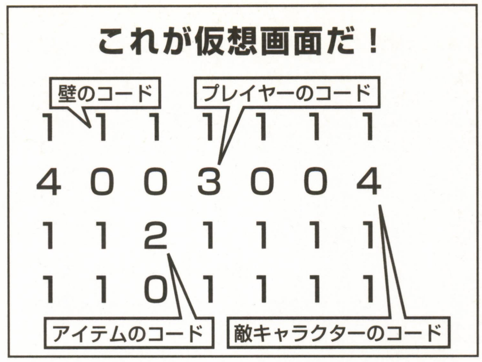
壁との衝突判断はどうする？

ゲーム・フィールドに壁／通路を取り入れる場合，登場するキャラクターは通路にそって移動させる必要があります。このためには，壁との衝突判定が必要です。

とはいえ，壁はゲーム・フィールドに多数存在しますね。シューティング・ゲームのときと同じように衝突判定をすると，処理に必要な時間



《第2図》キャラクター・アクション・ゲームでは，一般的にこのような画面表示をする



《第3図》仮想画面は，ゲーム・フィールドの状態に対応した数値が並べられたものだ

が長くなってしまいます。このため，キャラクター・アクション・ゲームに合った判定方法を考えなければなりません。

壁の判定方法の一つとして，“仮想画面”を用いる方法があります。

仮想画面を活用しよう！

仮想画面とは，ゲーム・フィールドの状態を格納する記憶領域のことで，具体的には配列変数を適用します。画面上の個々のブロックに，配列変数の要素一つ一つが対応するようにして，そのブロックの状態を配列変数の要素に数値データとして記録します。

たとえば，ゲーム・フィールドが**第2図**のような状態になっているときに，**第3図**のような数値データが記録されていれば，特定の座標の状況を数値データから判断することができるわけです。

2次元の配列変数を用意し，一つ目の添え字をX座標に，二つ目の添え字をY座標に対応させてやります。すると，座標から，そこに存在しているキャラクターのコードを即座に調べることができるのです。

仮想画面のデータはそのままでは変化しませんから，ゲーム開始時の初期化やキャラクターの移動の段階で，初期化したり更新してやる必要があります。このような処理を，キャラクター処理のプログラムに追加しておくことで，仮想画面を参照した衝突判定が行えるようになります。

キャラクターを移動させる場合，もともと表示されているキャラクターを消去し，新しい場所にキャラクターを表示しますね。仮想画面についても同様で，もともと存在した場所に対応する仮想画面のデータをクリアし，新しい場所にデータを書き込むという処理になります（**第4図**）。

このようにして，表示されている状態と完全に対応した仮想画面を用意してやれば，壁の判断は簡単なものになります。調べたい座標に対応した添え字を計算し，それによって仮想画面に対応した配列変数からデータを読みだすだけで済むわけです（**第5図**）。

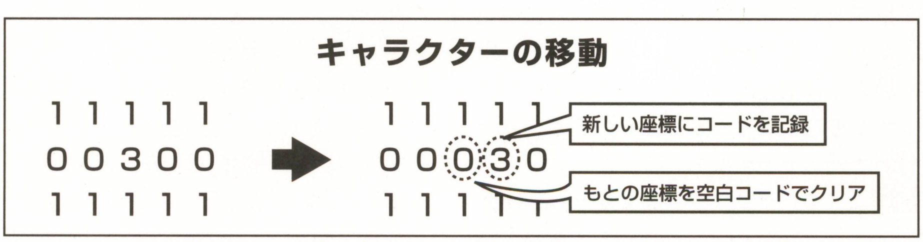
仮想画面を常時最新の状態に更新していく処理が余計にかかるものの，衝突判定が非常に簡略化されるため，トータルでは仮想画面を活用したほうが有利になるわけです。

キャラクターを移動させるには？

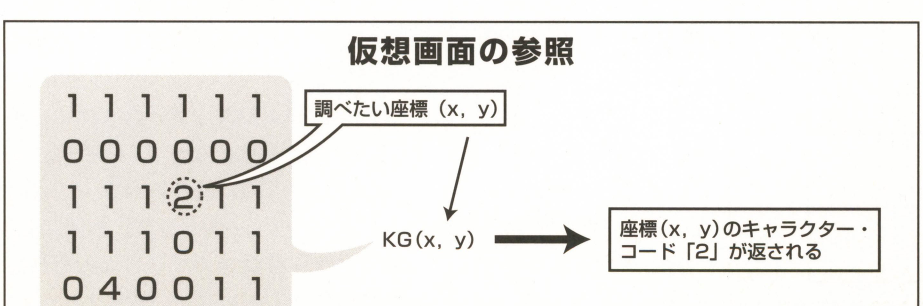
プレイヤー・キャラクターを移動させる場合は，壁によって進めない方向へ移動させないようにする必要があります。これは，仮想画面を参照すれば，壁の存在は簡単に判断できます。

移動方向に壁がある場合は移動させないようにすることで，プレイヤー・キャラクターは壁によって仕切られた通路を移動するようになります。また，敵キャラクターを移動させる場合も同様に，仮想画面を参照して通路部分を移動させることが可能です。

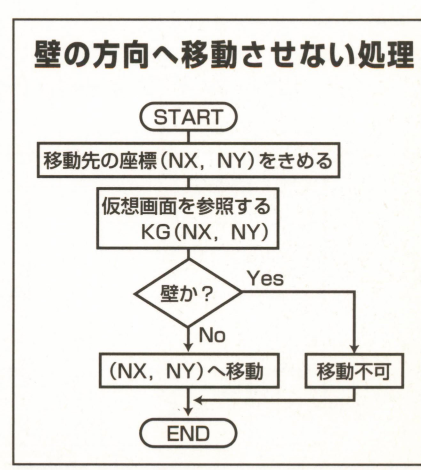
このフローチャートは，**第6図**のようになります。何らかの方法で移動後の座標を求め，その座標が壁かどうかを仮想画面によって判定し，壁であった場合は移動させないようにします。移動後の座標は，プレイヤー・キャラクターならばキー入力に応じてきめ，敵キャラクターならばランダムにきめたり何らかの移動アルゴリズムできめることになります。



《第4図》もといた位置を「0」にして，新しい位置に「3」を書き込めばOKだ

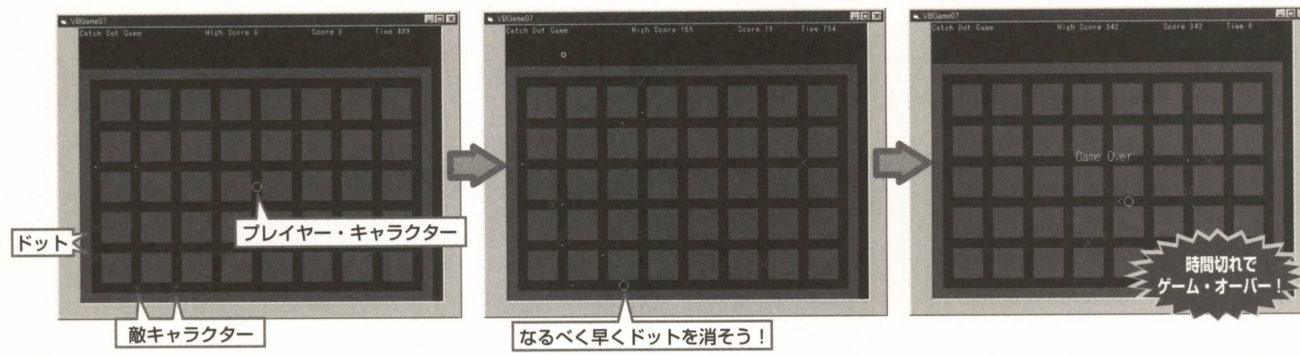


《第5図》仮想画面に対して座標を参照すれば，そこに何があるかも判別できる



《第6図》壁には移動できないように処理するフローチャート（流れ図）

これが『キャッチ・ドットゲーム』だ！



《第7図》敵キャラクターは，交差点で乱数によって向きを変えてくるため，あまり近づかないほうがよい。無理に接近してミスすると，時間をロスするぞ

壁であった場合に，その回の処理での移動をあきらめるか，あるいは移動できる方向を探して必ず移動させるようにするかは，ゲーム・デザインなどによって決定します。

『キャッチ・ドットゲーム』で遊ぼう！

それでは，実際にVBでプログラムされたキャラクター・アクション・ゲームで遊んでみましょう。プログラムは，**リスト1**です。

リストを入力して実行すると，**第7図**のような画面が表示されます。画面には，プレイヤーが操作するプレイヤー・キャラクターと，敵キャラクターが四つ登場します。敵キャラクターは，ランダムに格子状の通路を移動します。移動した後に，ドットを落としていく場合があります。

プレイヤーは，プレイヤー・キャラクターをテンキーの[2][4][6][8]で上下左右に移動させ，敵キャラクターが落としたドットを制限時間内になるべくたくさん拾い集めてください。

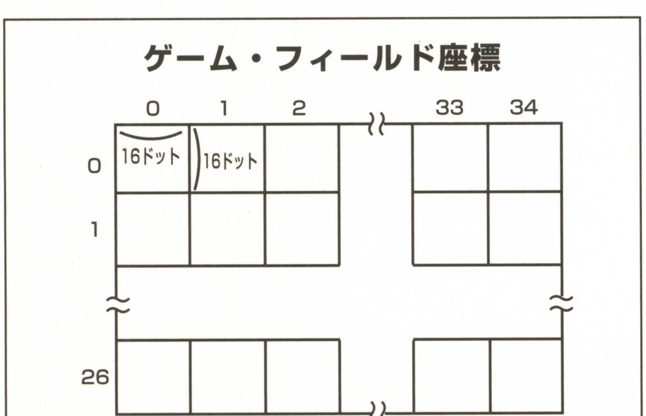
ドットを取ると，その段階での残り時間に応じた得点が加算されます。残り時間が多いほど，高得点になるわけです。ただし，敵キャラクターと接触すると，残り時間が大幅に減ってしまいます。

リストの処理を見てみよう

リスト1では，ゲーム・フィールドの座標を16ドット単位で構成し，キャラクターは16ドット単位で移動するようにしています。仮想画面は，16ドット単位のゲーム・フィールドでの座標系に対応しています（**第8図**）。

仮想画面には，壁の状況を記録するだけでなく，ドットの有無も記録しています。このゲームでは，壁とドットは同じ場所に存在することがあり得ないため，同じ仮想画面に記録することができるわけです。

プレイヤー・キャラクターは仮想画面でドットの有無を調べ，ドットがあった場合には得点アップなどの処理を行います。また，敵キャラクターは，キャラクター移動後にドットを描き直すかどうかをドットの有無できめています。これによって，「敵キャラクターが通過した際にドットを消してしまってドットの有無がわからなくなる」といったことが起こらないようになります。



《第8図》『キャッチ・ドットゲーム』は，このように16ドット単位のゲーム・フィールド座標で構成されている

通常，画面に登場するキャラクターは，対応する処理プログラムやデータ記録変数を持っていますが，ドットについては「ドット処理プログラム」も「ドット状態記録変数」もありません。仮想画面を活用することで，専用の処理プログラムや変数を使わずにキャラクターを扱っています。移動することがないキャラクターの場合は，このような仮想画面によるキャラクター処理が可能になる場合があるわけです。

敵キャラクターの移動は，格子状の通路の交差点の間を行き来するようなアルゴリズムになっています。敵キャラクターが交差点に存在する場合は，つぎの目標交差点を上下左右の方向の中から乱数で選びだします。仮想画面などで移動可能と判断された場合に，移動方向と移動カウンタを再設定しています。

これによって，交差点と交差点の間はスムーズに敵キャラクターが移動するようになっています。

まとめ

仮想画面を使うことで，ゲーム・フィールドの状況を簡単に参照できるようになることがわかったと思います。とくに，キャラクター・アクション・ゲームは仮想画面との相性がよく，仮想画面の使いこなしでプログラムを簡潔にまとめることができます。

今回のゲームでは，ゲーム・フィールドが完全な格子状になっているため，変化にとぼしいものになっています。次回は，ゲーム・フィールドを若干複雑にし，キャラクター移動アルゴリズムもそれに応じて高度なものにしていきます。

お楽しみに！

《リスト1》『キャッチ・ドットゲーム』のプログラム・リスト（VB6.0用）

```
Option Explicit

Dim keystat%(9) 'テンキーの状態を記録する変数
Dim keytrig0%, keytrig1%, keytrig2%, keytrig3% 'トリガー用キーの状態を記録す
る変数
Dim px%, py%, pf% 'プレイヤーキャラクタ
Dim ex%(3), ey%(3), evx%(3), evy%(3), ef%(3), ec%(3), ecc%(3) '敵キャラクタ
Dim kg(35, 26) '仮想画面
Dim hiscore%, score% 'ハイスコアとスコア
Dim ti% '残り時間
Dim dotc% 'ドットを発生させる間隔を数えるカウンタ
Dim gamestat% 'ゲーム進行状況

Private Sub Form_Load()
    hiscore = 0
    gameinit
End Sub

Private Sub gameinit() 'ゲーム初期化
    Dim ox%, oy%, x%, y%, i%
    With Form1
        .Width = Screen.TwipsPerPixelX * 640 'フォームのサイズを設定
        .Height = Screen.TwipsPerPixelY * 480
        .Caption = "VBGame07"
    End With
    With Picture1
        .Width = Screen.TwipsPerPixelX * 576 'ピクチャボックスのサイズを設定
        .Left = Screen.TwipsPerPixelY * 32
        .Height = Screen.TwipsPerPixelY * 432
        .Top = 0
        .ScaleMode = 3
        .ScaleHeight = 432
        .ScaleWidth = 576
        .BackColor = RGB(0, 0, 0)
        .ForeColor = RGB(255, 255, 255)
        .FillStyle = 0
        .Font = "ＭＳ ゴシック"
        .FontSize = 10
    End With
    Picture1.Cls
    Form1.Show

    Picture1.Print "Catch Dot Game"

    px = 17 'プレイヤーキャラクタの初期座標
    py = 15

    dotc = 100

    For i = 0 To 3
        ef(i) = 1 '存在フラグをたてる
        ec(i) = 3 '移動量
        ecc(i) = 0 '移動カウンタ
    Next

    ex(0) = 1 '敵キャラクタの初期座標
    ey(0) = 5
    ex(1) = 33
    ey(1) = 5
    ex(2) = 1
    ey(2) = 25
    ex(3) = 33
    ey(3) = 25

    '最初に壁でゲームフィールドを埋める
    For y = 4 To 26
        For x = 0 To 34
            kg(x, y) = 1 '壁に対応するコード
               Picture1.Line (x * 16, y * 16)-(x * 16 + 15, y * 16 + 15),
RGB(128, 0, 128), BF '壁表示
        Next
    Next

    '通路を作る
    For y = 5 To 25 Step 4
        For x = 1 To 33
            kg(x, y) = 0 '空白に対応するコード
               Picture1.Line (x * 16, y * 16)-(x * 16 + 15, y * 16 + 15),
RGB(0, 0, 0), BF '空白（通路）表示
        Next
    Next

    For x = 1 To 33 Step 4
        For y = 5 To 25
            kg(x, y) = 0 '空白に対応するコード
               Picture1.Line (x * 16, y * 16)-(x * 16 + 15, y * 16 + 15),
RGB(0, 0, 0), BF '空白（通路）表示
        Next
    Next
    score = 0
    ti = 1000
    putscore

    gamestat = 0 'ゲームモード（0でゲーム中、1でゲームオーバー状態）

    Timer1.Interval = 70 'インターバルタイマ起動
End Sub

Private Sub Picture1_MouseUp(button%, shift%, x!, y!)
    If (button And 2) = 2 Then 'マウス右ボタンクリックで強制終了
        End
    End If
End Sub

Private Sub Picture1_KeyDown(key%, shift%)
    If key = 27 Then 'ESCキーで強制終了
        End
    End If
    If key > 96 And key < 106 Then 'テンキー設定
        keystat(key - 96) = 1
    End If
    If key = 32 Then keytrig0 = 1 'トリガキー設定
    If key = 90 Then keytrig1 = 1
    If key = 88 Then keytrig2 = 1
    If key = 13 Then keytrig3 = 1
End Sub

Private Sub Picture1_KeyUp(key%, shift%)
    If key > 96 And key < 106 Then
       keystat(key - 96) = 0
    End If
    If key = 32 Then keytrig0 = 0
    If key = 90 Then keytrig1 = 0
    If key = 88 Then keytrig2 = 0
    If key = 13 Then keytrig3 = 0
End Sub

Private Sub Timer1_Timer()
    Dim npx%, npy%, dx%, dy%, nex%, ney%
    Dim i%, f!
    If gamestat = 0 Then '通常ゲームモード
        'プレイヤーキャラクタの処理
        npx = px
        npy = py
        If keystat(2) = 1 Then '下キーを押しているか？
            npy = py + 1
        ElseIf keystat(4) = 1 Then '左キーを押しているか？
            npx = px - 1
        ElseIf keystat(6) = 1 Then '右キーを押しているか？
            npx = px + 1
        ElseIf keystat(8) = 1 Then '上キーを押しているか？
            npy = py - 1
        End If
        If kg(npx, npy) <> 1 Then '移動先が壁以外か？
            putplayer px, py, 0
            putplayer npx, npy, RGB(255, 255, 255)
            px = npx '座標更新
            py = npy
            If kg(px, py) = 2 Then '移動先にドットがあるか？
                kg(px, py) = 0 '仮想画面のドットを消す
                putdot px, py, 0
                scoreup 10 + Int(ti / 100)    '得点アップ
            End If
        End If
        '敵キャラクタの処理
        For i = 0 To 3
            If ef(i) = 0 Then '存在していない場合
            ElseIf ef(i) = 1 Then '存在している場合
                 If px = ex(i) And py = ey(i) Then 'プレイヤーキャラクタとの
衝突判定
                    pf = 1 'プレイヤーダメージフラグ
                End If
                 If ecc(i) < 1 Then '移動カウンタが0になっている場合は新しい
移動方向を決める
                    f = Rnd(1) '乱数で新しい移動方向を決める
                     If f < 0.25 Then '乱数は0から1までなので0.25ずつに区切っ
て上下左右に対応させる
                        dx = 1
                        dy = 0
                    ElseIf f < 0.5 Then
                        dx = -1
                        dy = 0
                    ElseIf f < 0.75 Then
                        dx = 0
                        dy = 1
                    Else
                        dx = 0
                        dy = -1
                    End If
                    evx(i) = 0 '移動方向を一旦クリア
                    evy(i) = 0
                    nex = ex(i) + dx * 4 '次の交差点の座標
                    ney = ey(i) + dy * 4
                     If nex > 0 And nex < 35 And ney > 3 And ney < 26 Then '
ゲームフィールド範囲内か？
                        If kg(nex, ney) <> 1 Then '壁が無いか？
                            evx(i) = dx '移動できる場合は移動方向を新規設定
                            evy(i) = dy
                            ecc(i) = ec(i)
                        End If
                    End If
                Else
                    ecc(i) = ecc(i) - 1
                End If
                nex = ex(i) + evx(i) '移動先の座標
                ney = ey(i) + evy(i)
                putenemy ex(i), ey(i), RGB(0, 0, 0)
                putenemy nex, ney, RGB(255, 0, 0)
                If kg(ex(i), ey(i)) = 2 Then '元の場所にドットがあった場合
                    putdot ex(i), ey(i), RGB(255, 255, 0) 'ドットを描き直す
                End If
                 If dotc < 1 And Abs(dx) + Abs(dy) > 0 Then '一定間隔でドット
を発生させる
                    putdot ex(i), ey(i), RGB(255, 255, 0) 'ドットを描く
                    kg(ex(i), ey(i)) = 2 '仮想画面にドットを設定
                    dotc = 100
                End If
                dotc = dotc - 1 'ドット発生カウンタ更新
                ex(i) = nex
                ey(i) = ney
                 If px = ex(i) And py = ey(i) Then 'プレイヤーキャラクタとの
衝突判定
                    pf = 1 'プレイヤーダメージフラグ
                End If
            End If
        Next
        If pf > 0 Then 'プレイヤーがダメージを受けている？
            ti = ti - 200 '残り時間を減らす
            pf = 0
        End If
        ti = ti - 1
        If ti < 0 Then ti = 0
        puttime
        If ti < 1 Then gamestat = 1 '時間切れでゲームオーバー
    ElseIf gamestat = 1 Then
        With Picture1
            .CurrentX = 230
            .CurrentY = 200
            .FontSize = 15
            .ForeColor = RGB(Rnd * 255, Rnd * 255, Rnd * 255)
        End With
        Picture1.Print "Game Over"
        If keytrig3 = 1 Then
            Timer1.Interval = 0 'インターバルタイマ停止
            gameinit
        End If
    End If
End Sub

Private Sub putscore()
    Picture1.Line (200, 0)-(369, 15), 0, BF
    With Picture1
        .CurrentX = 200
        .CurrentY = 0
        .FontSize = 10
        .ForeColor = RGB(255, 255, 255)
    End With
    Picture1.Print "High Score " & hiscore
    Picture1.Line (370, 0)-(469, 15), 0, BF
    With Picture1
        .CurrentX = 370
        .CurrentY = 0
        .FontSize = 10
        .ForeColor = RGB(255, 255, 255)
    End With
    Picture1.Print "Score " & score
End Sub

Private Sub puttime()
    Picture1.Line (470, 0)-(575, 15), 0, BF
    With Picture1
        .CurrentX = 470
        .CurrentY = 0
        .FontSize = 10
        .ForeColor = RGB(255, 255, 255)
    End With
    Picture1.Print "Time " & ti
End Sub

Private Sub scoreup(up)
    score = score + up
    If hiscore < score Then hiscore = score
    putscore
End Sub

Private Sub putplayer(x, y, c)
    Picture1.Circle (x * 16 + 8, y * 16 + 8), 7, c
End Sub

Private Sub putdot(x, y, c)
    Picture1.Line (x * 16 + 7, y * 16 + 7)-(x * 16 + 8, y * 16 + 8), c, BF
End Sub

Private Sub putenemy(x, y, c)
    Picture1.Line (x * 16, y * 16)-(x * 16 + 15, y * 16 + 15), c
    Picture1.Line (x * 16 + 15, y * 16)-(x * 16, y * 16 + 15), c
End Sub
```

《第1表》『キャッチ・ドットゲーム』の主な変数

変数名	内容
hiscore	ハイスコア
score	プレイヤー・スコア
px	プレイヤー・キャラクターX座標
py	プレイヤー・キャラクターY座標
ef()	敵キャラクター状態フラグ
ex()	敵キャラクターX座標
ey()	敵キャラクターY座標
evx()	敵キャラクターのX方向の進行方向（－1，0，1のいずれか）
evy()	敵キャラクターのY方向の進行方向（－1，0，1のいずれか）
ec()	敵キャラクター進路変更間隔
ecc()	敵キャラクター進路変更間隔カウンタ（カウント・ダウンするカウンタで，交差点にきたときに0になるような初期値を設定する）
kg()	仮想画面用2次元配列変数（0：空白，1：壁，2：ドット）
ti	残り時間
dotc	ドットを発生させる間隔計測カウンタ
gamestat	ゲーム進行状況（0：ゲーム中，1：ゲーム・オーバー）

リスト1入力&実行方法

①VB起動

まずは，VBを起動します。新しいプロジェクトとして，「標準EXE」を選択して[開く]ボタンをクリックしてください。通常は，最初に標準EXEが選択された状態になっているので，[開く]をクリックするだけです。すると，**写真①**のような画面が表示されます。

②フォームの設定

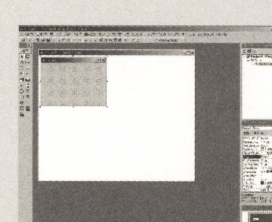
写真②のように，フォームを設定してください。プロパティは，プログラム中で更新されるので設定不要です（不用意に設定すると動作しなくなる場合があるので注意）。設定するのは，PictureBoxとTimerです。それぞれツールボックスのアイコンを選択した後に，フォーム上に貼り付けてください。貼り付ける操作は，フォーム上で左クリックした後，右下方向に少しドラッグしてから左ボタンを離せばOKです。位置，サイズなどは自由です（プログラム中で修正される）。

③リストの入力

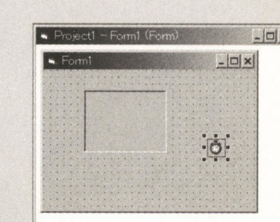
プロジェクト・ウィンドウのコード表示ボタンをクリックすると，コード・ウィンドウが表示されます。Form1のコードとして，**リスト1**を入力してください。

④実行

実行メニューから[開始]を選択すると，プログラムが実行されます。

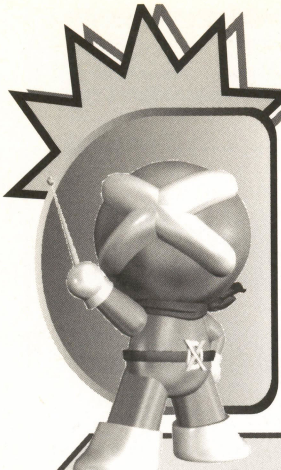


《写真①》VBを起動したら「標準EXE」を選ぼう



《写真②》PictureBoxとTimerをフォームに配置する

マイクロソフトの技術者がDirectXテクノロジーを伝授する！

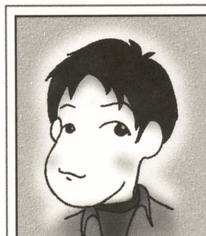


▲X君

IE+DirectAnimation マルチメディア・プログラミング

第7回　マウスのイベント処理テクニック

TEXT&PROGRAMING●平井昌人



横浜市生まれの＆H25歳。1993年4月に開発言語製品のサポート・エンジニアとしてマイクロソフトに入社。現在は，システム・エンジニア（SE）としてVisual Studioなどの開発ツール製品やテクノロジーの啓蒙／支援活動を行っている。この秋は，MSからWindows XPやX-Boxなどが発売される。X-Boxは，ぜひともがんばってほしいものだ。

このコーナーでは，『Microsoft Internet Explorer（マイクロソフト インターネット エクスプローラ）(以下IEと略)』と『DirectAnimation（ダイレクトアニメーション）*1』を使って，誰でも簡単に作れるマルチメディア・プログラムを紹介します。先月号は，連続した静止画像を高速に切り替えて，キャラクターの複雑な動きや変化を表現する2Dアニメーションを紹介しました。今回は，マウスのイベントに反応させるという新しいテクニックについて解説します。

＊1　DirectAnimation
2Dや3Dグラフィックス，サウンド，映像などを総合的にサポートする高レベルなAPIを提供している。IEに統合されており，インターネットでの利用を強く意識した設計となっている。

＊2　イベント
Windows上で起こる，あらゆる“出来事”を総称していう。マウスが動いた，マウスがクリックされた，キーが押された／離された，指定した時間が経過した，などが代表的なイベント。他にも，ウィンドウが閉じたとかメールが着信したなどの事象もイベントといえる。サンプル・コードにある「Sub Window_OnLoad()」や「Sub Window_OnResize()」などをイベント・プロシージャといい，特定のイベント（ロードされた／ウィンドウのサイズが変わった）が発生すると，そのプロシージャが実行される。

今回作るサンプルはコレ！

これまで紹介してきたサンプルは，すべてプログラムの中で動きや変化を定義してアニメーションさせていました。しかし，イベント*2を使うことで，ユーザーの操作に反応する複雑で多彩な動きや変化を組み込むことができるようになります。

今回は，マウスのイベント処理に注目して，**写真1**のようなサンプルを作成してみましょう。クリックした位置に，効果音とともにキャラクターが現れるというインタラクティブな（相互に作用する）アニメーションです。

イベントを使うと何ができる？

一般的に利用されるイベントとしては，

●マウス

●キーボード

●時間経過

があります。

DirectAnimationでイベントを利用すると，いったい何ができるようになるのでしょうか？イベントの用途は幅広く，いろいろと応用することができます。いくつか実例をあげると，

●マウスのクリックで色や動きを変える

●クリックした位置にアクションを起こす

●キー操作でキャラクターを動かしたり弾を発射させたりする

●指定した時間が経つとキャラクターが現れる／消える

などいろいろとあります。

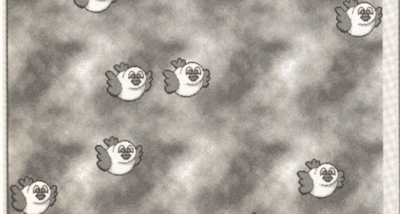
今回は，マウス・ポインタの座標値の取得と，マウスのクリック・イベントの処理について紹介しましょう。

マウス・ポインタの座標値を取得する方法は？

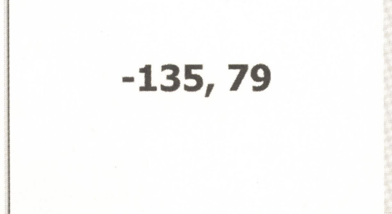
イベントとは直接関係ありませんが，はじめにマウス・ポインタ（マウス・カーソル）の位置を取得する方法を覚えておきましょう。

サンプル・プログラム『マウス・ポインタの座標を取得するサンプル』をベースに解説していきます。ソースコードは，**リスト1**です。実行画面は，**写真2**のようになります。

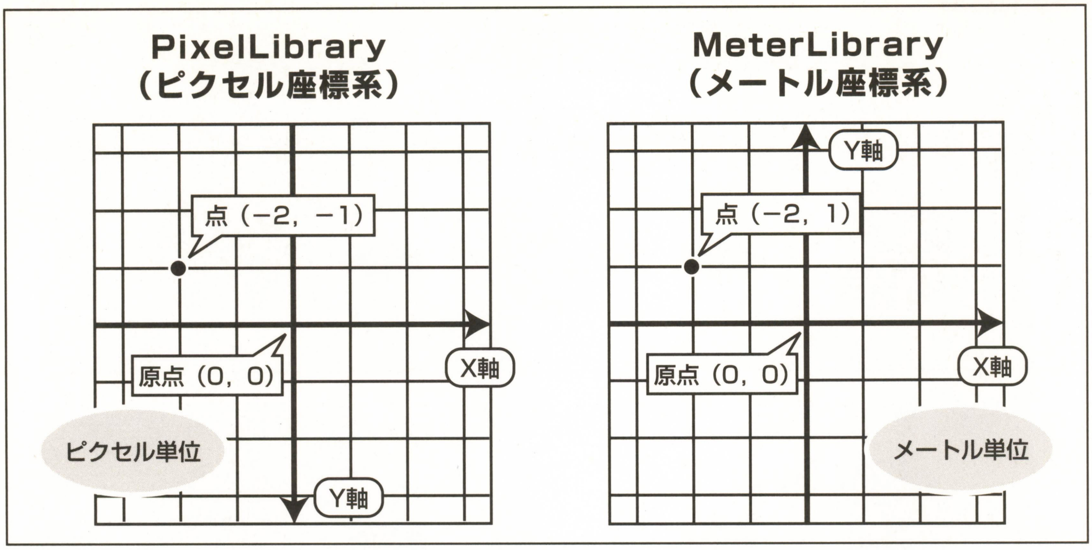
このプログラムは，マウス・ポインタのXY座標の値を画面上に表示します。マウスを動かすと，表示されている値も変化します。



《写真1》今回作成するサンプル『フライング・バード』。かわいいでしょ



《写真2》『マウス・ポインタの座標』を実行すると，座標がリアルタイムに表示される



《第1図》DirectAnimationの座標系は，ピクセル座標系とメートル座標系の二つがある

マウス・ポインタの座標値を取得～MousePositionプロパティ

マウス・ポインタの座標値は，DAStaticsオブジェクトのMousePositionプロパティを参照すればわかります。しかし，MousePositionプロパティは，DAPoint2オブジェクト*3を返します。

つぎのコードは，マウス・ポインタの座標値をDAPoint2オブジェクト変数（mouse）に格納します。

```
Set mouse = m.MousePosition
```

X座標値とY座標値の抽出～Xプロパティ，Yプロパティ

MousePositionプロパティで取得したDAPoint2オブジェクトから，実際のX座標の値とY座標の値を抽出するためには，DAPoint2オブジェクトのXプロパティとYプロパティを参照します。XプロパティやYプロパティで取得した値は，DANumberオブジェクトなので注意してください。

つぎのコードは，DAPoint2オブジェクト（mouse）のXプロパティを参照して，マウス・ポインタのX座標値をDANumberオブジェクトの変数（numX）に格納します。

```
Set numX = mouse.X
```

メートル→ピクセル変換～Pixelプロパティ

じつは，XプロパティとYプロパティで取得した値は，メートル単位となっています。第1図のように，プログラムで使用している座標系ライブラリ（ピクセル座標系*4とメートル座標系*5）にかかわらず，つねにメートル座標系（MeterLibrary）として報告されます。

このサンプルは，ピクセル座標系（PixelLibrary）を利用しているため，メートル値をピクセル値に変換しなければなりません。メートルからピクセルへの変換は，つぎのようにDivメソッドを使い，メートル値をm.Pixelで除算します。

```
Set numX = m.Div(numX, m.Pixel)
```

また，メートル座標系とピクセル座標系では，Y軸のプラス／マイナスが反対になっているため，さらにMulメソッドを使いY座標値に−1を掛け合わせて反転させておきます。

```
Set  numY  =  m.Mul(m.Div(numY,
m.Pixel), m.DANumber(-1))
```

DANumberオブジェクトからDAStringオブジェクトに変換～ToStringメソッド

このサンプルでは，XとYの座標値を画面に表示させています。画面に表示させるためには，イメージ（DAImageオブジェクト）を作りださなければなりませんが，DANumberオブジェクトを直接イメージ（DAImageオブジェクト）に変換することはできません。

このため，つぎのようにToStringメソッド*6でDANumberオブジェクト（numX）をDAStringオブジェクト（strX）に変換します。

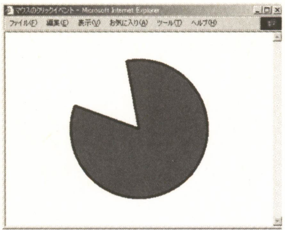
```
Set strX = numX.ToString(0)
```

*3　DAPoint2オブジェクト
DirectAnimationでは，平面（2次元）座標上の点（x，y）をDAPoint2オブジェクトで管理している。Point2メソッドを使って，X，Yの座標値からDAPoint2オブジェクトを作成できる。たとえば，点（4，5）は「m.Point2(4, 5)」という式となる。逆に，DAPoint2オブジェクトのXプロパティ，Yプロパティを参照すれば，それぞれX座標，Y座標の値が取得できる。

*4　ピクセル座標系
座標の単位がピクセル（Pixel）になる。X軸は右方向がプラス，Y軸は下方向がプラスとなっている。DAControlのPixelLibraryプロパティで初期化する。ピクセル座標系は，Windowsと同じ単位なので2Dグラフィックスを利用する場合に便利。

*5　メートル座標系
座標の単位がメートル（m）になる。X軸は右方向がプラス，Y軸は上方向がプラスとなっている。DAControlのMeterLibraryプロパティで初期化する。メートル座標系は，主に3Dを取り扱う場合に利用される。

*6　ToStringメソッド
DANumberオブジェクトをDAStringオブジェクトへ変換するメソッド。VBやVBScriptでのCStr関数やFormat関数に似ている。ToStringメソッドの引数は，小数点以下何桁まで必要なのかをあたえる。たとえばDANumberオブジェクト型の変数numを小数点以下3桁までの文字列（DAStringオブジェクト）へ変換するには，num.ToString(3)と記述すればよい。



《写真3》サンプル『マウスのクリック・イベント』の実行画面

＊7　DAStringメソッド
String型の文字列を，DAStringオブジェクトに変換するメソッド。たとえば"ABC"をDATringオブジェクトにする場合は，「m.DAString("ABC")」と記述する。DirectAnimationのプログラミングの世界では，数値はDANumberオブジェクト，文字列はDAStringオブジェクトで取り扱っているのだ。

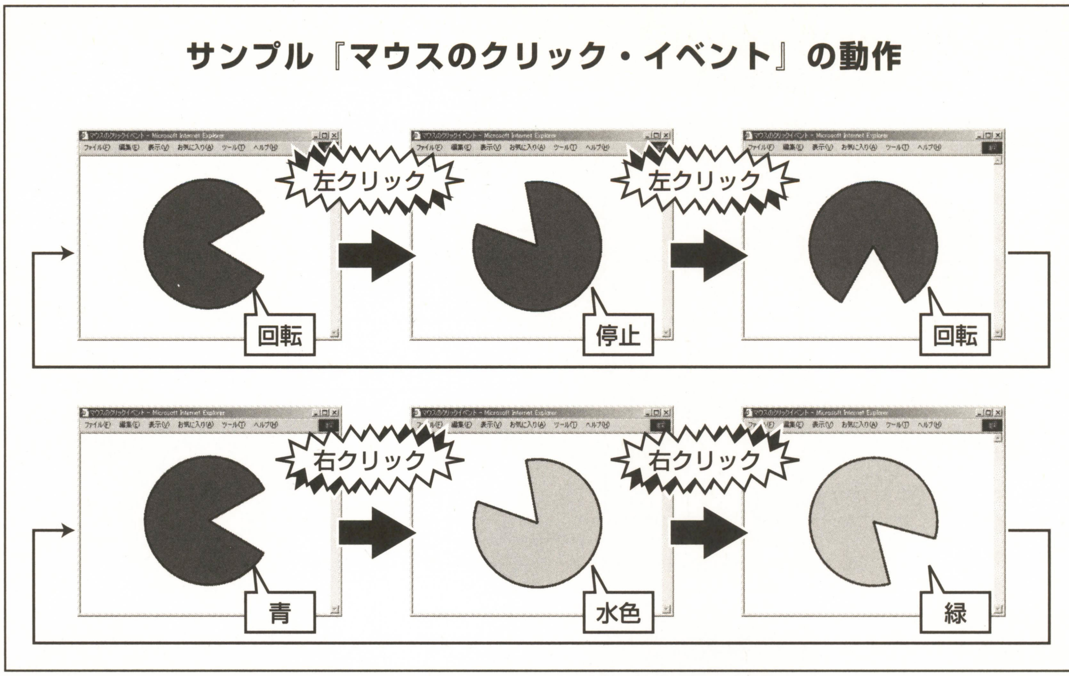
＊8　マウスのイベント
Windowsでは"クリック"とか"ダブル・クリック"とかいう操作が定義されているが，マウスという装置ではこの考えかたをしない。マウスのボタンが押された／離されたという単純なイベントとしてしか認識しないのだ。つまり，クリックはマウスの左ボタンが押された(m.LeftButtonDown)というイベントに等しくなる。

特別記念品付き
DirectAnimation
プログラム投稿受付中！

みなさんからのDirectAnimation／DirectX for Visual Basicプログラムの投稿をお待ちしております。投稿されたプログラムは，マイクロソフト平井さんに審査していただき，優秀作品は特選投稿プログラムコーナーに掲載されます。なんと，規定原稿料のほかに，マイクロソフト社から提供される『Visual Studio 97オリジナルTシャツ』をプレゼントします！どしどしご応募ください。

■投稿要項
DirectAnimationを使ったプログラムは，できるだけVBScriptで書かれたものが対象です（どうしてもできない場合はVBも可）。DirectX for Visual Basicは，VBのソースおよびexeファイルを同送してください。画像，サウンドなどは，著作権および肖像権などを侵害しないように注意しましょう。
投稿のきまりは，特選プログラムコーナーに準じますが，電子メールでも受け付けます。

■あて先
〒141-8715東京都品川区東五反田1-11-15電波新聞社マイコンBASICマガジン「特選プログラムコーナーDirect Animation」係
電子メール：webmas@basicmagazine.net



《第2図》このサンプルでは，左クリックと右クリックでちがう処理が行われる

文字列(DAStringオブジェクト)の連結～ConcatStringメソッド

VBScriptやVBでは，文字列の連結を「"ABC" + "XYZ"」や「"ABC" & "XYZ"」のように記述します。しかし，DirectAnimationのDAStringオブジェクトの連結は，ConcatStringメソッドを使わなければなりません。

また，文字列そのものは，DAStringメソッド*7で変換しておく必要があります。

```
Set strPos = m.ConcatString(strX,
m.DAString(", "))
```

文字列からイメージを作成～StringImageAnimメソッド

文字列からその文字列が描画されてイメージを作成するのは，すでに紹介したStringImageメソッドまたはStringImageAnimメソッドを使います。

このサンプルでは，変動する文字列（DAStringオブジェクト）を使うので，StringImageAnimメソッドのほうを利用します。

```
Set img = m.StringImageAnim(strPos,
fnt)
```

マウスのイベント処理の手順

それでは，さっそくマウスのイベントを処理するテクニックを紹介しましょう。

ここでは，サンプル・プログラム『マウスのクリック・イベント』をベースに解説していきます。ソースコードは，**リスト2**です。実行画面は，**写真3**のように画面中央で青色の扇形のイメージが回転しています。そして，**第2図**のように二つのイベントに対応してアニメーションが変化します。

●マウスの左ボタンを押す(クリックする)
扇形の回転がストップ。再度クリックすると，回転がスタートします。

●マウスの右ボタンを押す(右クリックする)
扇形の色が，「青→水色→緑」と変化します。

DirectAnimationが提供しているマウスのイベント

DirectAnimationでは，プログラムでイベントを簡単に取り扱うために，DAEventオブジェクトが用意されています。

第1表は，よく利用するマウスのイベント*8，キーボードのイベント，タイマーイベントです。それぞれ，DAEventオブジェクトを返します。

《第1表》よく利用されるマウス／キーボード／タイマーのイベント

イベント名	内容
m.LeftButtonDown	マウスの左ボタンが押された
m.LeftButtonUp	マウスの左ボタンが離された
m.RightButtonDown	マウスの右ボタンが押された
m.RightButtonUp	マウスの右ボタンが離された
m.KeyDown(keyCode)	keyCodeで指定されるキーが押された
m.KeyUp(keyCode)	keyCodeで指定されるキーが離された
m.Timer(timeout)	timeoutで指定された秒数が経過した

マウスのクリック（左ボタンを押す）に対応させる方法〜Untilメソッド

DirectAnimationが提供しているほとんどのオブジェクトは，イベントに対応させてその内容を変化させることができます。

基本的な考えかたは，「はじめは○○で，××したら△△に変える」となります。○○と△△がDirectAnimationの同種のオブジェクトで，××がイベントとなります。

つぎのコードは，Untilメソッドを使って「はじめは青色（m.Blue）でマウスがクリック（m.LeftButtonDown）されたら水色（m.Cyan）に変化させる」というイベントに対応したDAColorオブジェクト（col）を作ります。

```
Set col = m.Until(m.Blue, m.Right
ButtonDown, m.Cyan)
```

このDAColorオブジェクト（col）を背景の色やイメージの色に利用するのです。

イベントによる変化を繰り返すためのテクニック〜Initメソッド

さて，上記の方法でイベントに対応するオブジェクトが作成できましたが，これでは1回クリックしたら青から水色に変わってそれで終わりです。再度クリックしても，もう何も変化しません。

そこで，もう一度クリックしたら青にもどるように，変化を繰り返すように手直ししたいと思います。この手法は，今までにない変わったコーディングのスタイルとなります。

まず，変化の対象となるオブジェクト（ここではDAColorオブジェクト）を，つぎのようにCreateObjectメソッドで新しいインスタンスとして定義します。

```
Set col = CreateObject("Direct
Animation.DAColor")
```

CreateObjectメソッドの引数が，変化の対象とするDirectAnimationのオブジェクトです。DANumberオブジェクトであれば“DirectAnimation.DANumber”，DAImageオブジェクトであれば“DirectAnimation.DAImage”，DASoundオブジェクトであれば“DirectAnimation.DASound”というように文字列であたえます。

つぎに，InitメソッドとUntilメソッドを使って，そのオブジェクトを初期化します。コードは多少複雑ですが，ポイントは，Untilメソッドがネストしている（入れ子になっている）ことと，最後に指定するオブジェクトがインスタンス化[*9]したオブジェクト変数であることです。

つぎのコードは，右クリックで青（m.Blue）→水色（m.Cyan）→緑（m.Lime）を繰り返します。

```
col.Init m.Until(m.Blue, m.Right
ButtonDown, _
         m.Until(m.Cyan, m.Right
ButtonDown, _
         m.Until(m.Lime, m.Right
ButtonDown, col)))
```

サンプルでは，この色（col）を扇形の塗りつぶしに利用しています。

動作の停止／再開を行うためのテクニック

このサンプルには，もう一つイベントが組み込まれています。クリックしたときに扇形の回転が止まり，再度クリックするとまた回転をはじめるという動作です。

このような動作は，アニメーションの再生の進行時間を制御しているSubstituteTimeメソッド[*10]で処理することができます。デフォルトでは，時刻と同じスピードです。

サンプルでは，扇形のイメージが毎秒180度で回転していますが，このイメージのSubstituteTimeメソッドとIntegralメソッド[*11]を使ってつぎのように処理すると，進行時間が2倍になり，結果として回転スピードも2倍になります。

```
Set img = img.SubstituteTime(m.
Integral(m.DANumber(2)))
```

そして，Integralメソッドに0を渡せば，進行時間が止まるので回転も停止します。サンプルでは，このメカニズムを利用して回転の停止／再開を実現しています。

サンプル・プログラム『フライング・バード』を実行してみよう！

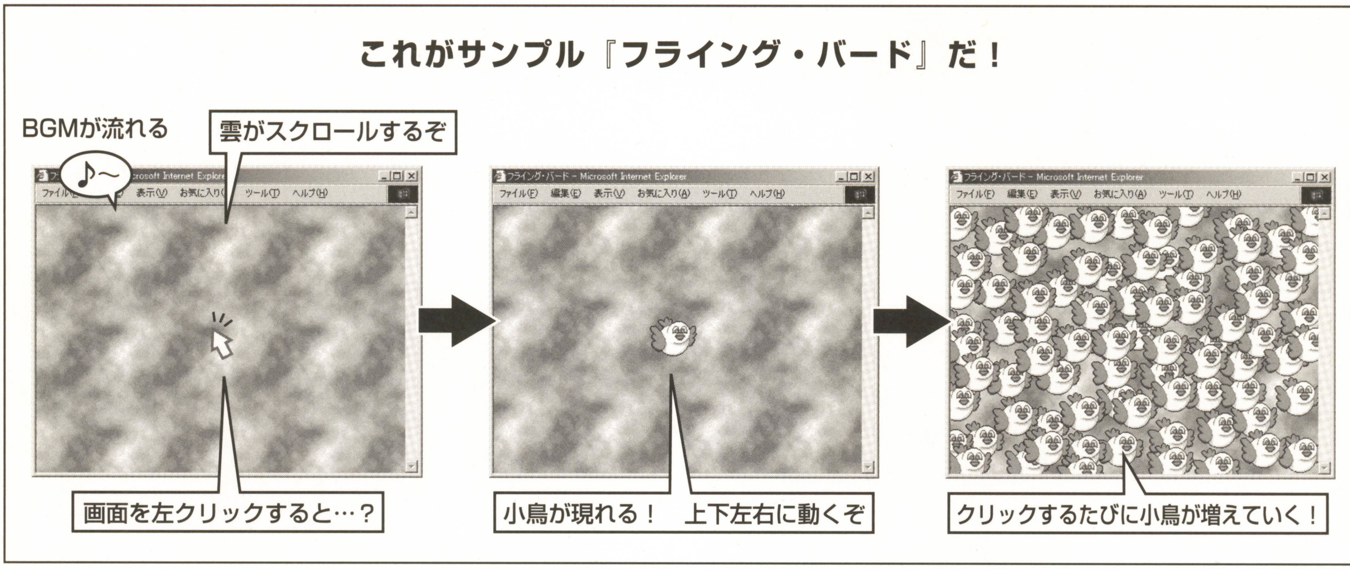
今回のテーマである，マウス・ポインタ座標とクリック・イベントのテクニックを駆使したサンプル・プログラム『フライング・バード』“FlyBird.htm”を用意しました。

ソースコードは，**リスト3**です。なお，サンプルのソースコードや画像ファイル，サウンド・

＊9　インスタンス化
インスタンスは，オブジェクトがメモリ上に確保され実態となっていることをいう。インスタンス化するとは，そのオブジェクトを空っぽの状態で生成するということ。たとえば，DirectAnimation.DAColorをCreateObjectメソッドでインスタンス化すると，色がまだ定義されていない未初期化のDAColorオブジェクトが作成できる。Initメソッドは，このような初期化されていないオブジェクトに対して一度だけ使うことができる。

＊10　SubstituteTimeメソッド
オブジェクト単位でアニメーションの進行時間を管理するメソッド。このメソッドを使えば，そのオブジェクトのアニメーションの進行時間を自由に変更することができる。通常は，スタートしてから時刻と同じ進み具合で処理されているが，変動値を利用してスピード・アップしたり，スローダウンさせたり，止めたりと，完全に制御することが可能になる。IntegralメソッドやInterpolateメソッドで定義した変動値を利用することが多い。「SubstituteTime(m.Integral(m.DANumber(1)))」は通常のスピード，「SubstituteTime(m.Integral(m.DANumber(0.5)))」は半分のスピード，「SubstituteTime(m.Integral(m.DANumber(2))」は倍速となる。

＊11　Integralメソッド
引数で指定されたレートにしたがったタイマー・カウンタを返すメソッド。「Integral(m.DANumber(1))」はm.LocaTimeと同じで，通常の時刻と同じスピード。「Integral(m.DANumber(2))」は，通常の2倍のスピードで時刻が刻まれる。つまり，時間の進みぐあいが標準の2倍となるのだ。



《第3図》クリックするたびに小鳥が増えていくぞ。おもしろいから，ぜひ実行してみよう！

ファイルなどは，ベーマガ・ホームページにありますのでダウンロードしてください。

●http://www.basicmagazine.net/

実行すると，**第3図**のように軽いBGMとともに，雲がスクロールしている画面が表示されます。画面の適当な位置を左クリックすると，クリックした位置にかわいい小鳥が現れ，フワフワ飛んで行きます。

小鳥は，クリックするたびにその位置に現れます。画面をつぎつぎとクリックしていくと，**第3図**右側のように小鳥が増えていきます。また，クリックしたときには効果音も鳴ります。

サンプルで使用しているイメージ・ファイル／サウンド・ファイルは，つぎのとおりです。

●CH223.GIF*12

キャラクターとして使うアニメーションGIFファイルです（**写真4**）。

●BCG_024.GIF*13

背景として使うGIFファイルです（**写真5**）。

●bfire.wav

効果音として使う，0.17秒の短いウェーブ・ファイルです。

●GM091.MID*14

BGMとして使うMIDIファイルです。



《写真4》アニメーションGIFファイル“CH223.GIF”



《写真5》背景のGIFファイル“BCG_024.GIF”

アニメーションGIFは便利！

DirectAnimationでは，アニメーションGIF*15をイメージ（DAImage）としてロードし表示すると，本来設定されているとおりのアニメーションがきちんと再生されます。

先月号では，何枚もの静止画像を連続表示させてキャラクターの複雑な動きを表現していました。しかし，このようにあらかじめアニメーションGIFとして準備しておけば，プログラミングがとてもスマートになります。

さらに動きを追加する！

サンプルの小鳥のアニメーションGIFは，鳥が羽ばたくアニメーションです。

しかし，本来のアニメーションGIFの動きでは小さすぎるため，DirectAnimation側でさらに上下左右のフワフワとした動きを追加します。その動きは，X軸方向（左右）に±32ピクセル，Y軸方向（上下）に±48ピクセルぶん変位する2D変換です。

クリックした位置にキャラクターを移動〜Translate2Pointメソッド

クリックした位置に小鳥が表示されるのは，小鳥のイメージをクリックした時点のマウスの位置へと移動させているからです。

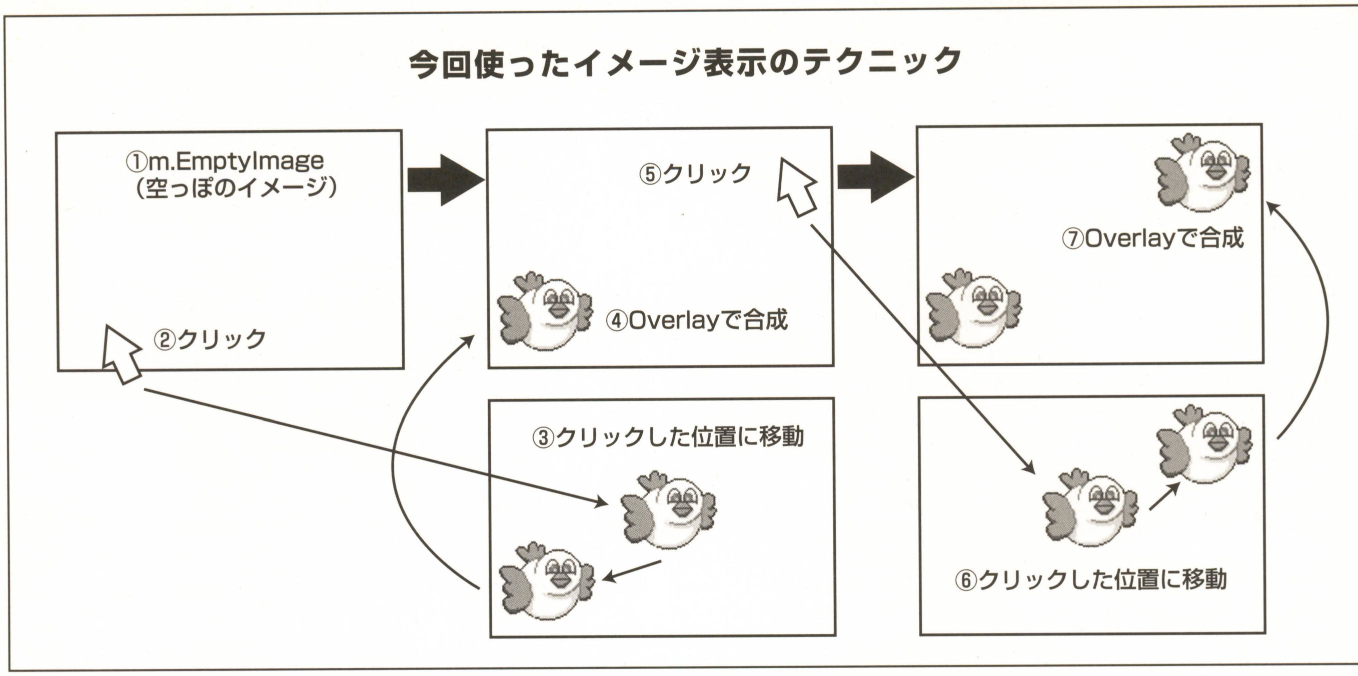
マウスの位置は，MousePositionプロパティでDAPoint2オブジェクトとして取得します。し

*12　CH223.GIF
このアニメーションGIFファイル“CH223.GIF”は，ナツメ社『インターネットホームページ素材集Pack3000 GIFアニメーション＆サウンド編』から使用しています。

*13　BCG_024.GIF
このGIFファイル“BCC_024.GIF”は，株式会社インプレス『Hybrid版Internetホームページ素材集Cool＆ビジネス編』から使用しています。データの著作権は，株式会社インプレスに帰属します。

*14　GM091.MID
サンプルのBGMに使用しているMIDIファイル“GM091.MID”は，有限会社キュービックエナジー発売の『MusicEnergy5』から使用しています（1998 Cubic Energy Inc. All rights reserved.）。

*15　GIF／アニメーションGIF
GIFは，【Graphics Interchange Format】の略。グラフィック・データの画像フォーマットの一種で，アメリカのCompuServe社で開発された。モノクロは256階調，カラーは256色まで扱える。JPEGと並んで，インターネットの世界でよく使われる。また，連続した何枚ものGIF画像を順に表示することでアニメーションを実現しているものを，アニメーションGIFと呼んでいる。



《第4図》イメージが独立して重なっていくメカニズムは，このようになっている！

かしこの値は，マウスの動きに合わせてつねに変化してしまいます。そのため，つぎのコードのように，クリックした時点の値に固定させなければなりません。

```
Set mouse = m.UntilEx(mouse,
m.Always.Snapshot(mouse))
```

そして，Translate2Pointメソッド*16でその位置へ移動させます。

```
Set chra = chra.Transform(m.Trans
late2Point(mouse))
```

イメージが独立して重なっていくメカニズム

このサンプルのアニメーションは，ちょっとしたテクニックを使っています。

画面をクリックするたびに，小鳥のイメージが増えていきますね。プログラムでは，一つのイメージしか作成していません。

この仕組みは，**第4図**のように，クリックしたときに既存のイメージに小鳥のイメージを合成（Cverlay）していくのです。つまり，はじめは空っぽの透明なイメージ（EmptyImageプロパティ）を用意しておき，クリックするたびにその上に新しく小鳥のイメージを重ねていきます。

```
birds.Init m.Until(m.EmptyImage,
m.LeftButtonDown, m.Overlay(birds,
chra)))
```

クリック時に効果音をだすには？

マウスをクリックしたときに効果音が鳴るように，サウンド（DASoundオブジェクト）にもイベントを利用します。

サンプルでは，つぎのコードのように，何もしないときは無音（Silenceプロパティ*17）で，クリックすると“bfire.wav”の再生を行います。0.18秒経過（Timerメソッド）すると，もとにもどるように仕込んであります。

```
eff.Init m.Until(m.Silence, m.LeftButtonDown, _
        m.Until(m.ImportSound("bfire.wav").Sound,
                m.Timer(0.18), eff))
```

まとめ

今回は，DirectAnimationにイベントを組み込んで，インタラクティブな動きをするプログラミングを紹介しました。

イベントを利用することで，かなりゲーム色がでてくることがわかると思います。多少高度で難解な部分もありますが，『Microsoft（マイクロソフト） Visual Basic（ビジュアルベーシック）』や『Microsoft（マイクロソフト） Visual（ビジュアル） C＋＋（シープラスプラス）』のような他のプログラミング言語でここまで簡素には作れません。まさに，DirectAnimationならではの機能といえます。

みなさんも，ぜひ挑戦してみてくださいね。それでは，また来月！

＊16　Translate2Pointメソッド
DAPoint2オブジェクトで指定された点（x，y）への移動を定義するメソッドで，DATransform2オブジェクトを返す。これに似たようなメソッドにTranslate2メソッドがあるが，これは現在の位置からX方向にいくつ，Y方向にいくつ移動するという相対的な移動を定義する。「m.Translate2Point (m.Point2(3, 4))」は点（3, 4）への移動を，「m.Translate2 (3, 4)」はX方向に+3，Y方向に+4動くという移動を意味する。

＊17　Silenceプロパティ
無音状態のDASoundオブジェクトを返すプロパティ。無音＝音がしない状態が必要なときは，このプロパティから取得したDASoundオブジェクトを利用する。

《リスト1》サンプル『マウス・ポインタの座標』"mosMove.htm"（VBScript用）

```
<HTML>
<TITLE>マウスポインタの座標</TITLE>
<BODY LEFTMARGIN = "0", TOPMARGIN = "0">
<SCRIPT LANGUAGE = "VBScript">
'//--------------------------------------------------------------
'//
'// DirectAnimation Sample                        by Masato Hirai
'//
'//    『マウスポインタの座標』
'//
'//--------------------------------------------------------------
Option Explicit

'//--------------------------------------------------------------
'// 初期化処理
'//--------------------------------------------------------------
Sub Window_onLoad()
  '//
  '// 変数・定数の宣言
  '//
  Dim m       'As DAStatics    'DAStaticsオブジェクト
  Dim mouse   'As DAPoint2     'マウスポインタの座標(x, y)
  Dim numX    'As DANumber     'X座標(DANumberオブジェクト)
  Dim numY    'As DANumber     'Y座標(DANumberオブジェクト)
  Dim strX    'As DAString     'X座標(DAStringオブジェクト)
  Dim strY    'As DAString     'Y座標(DAStringオブジェクト)
  Dim strPos  'As DAString     '表示用文字列(x, y形式)
  Dim fnt     'As DAFontStyle  'フォント
  Dim img     'As DAImage      'イメージ
  '//
  '// DirectAnimationの初期化（ピクセル座標系）
  '//
  Set m = DAControl.PixelLibrary
  '//
  '// マウスポインタの座標を取得
  '//
  Set mouse = m.MousePosition
  '//
  '// X座標値とY座標値を抽出
  '//
  Set numX = mouse.X
  Set numY = mouse.Y
  '//
  '// メートル単位からピクセル単位へ変換
  '//
  Set numX = m.Div(numX, m.Pixel)
  Set numY = m.Mul(m.Div(numY, m.Pixel), m.DANumber(-1))
  '//
  '// DANumberオブジェクトからDAStringオブジェクトへ変換
  '//
  Set strX = numX.ToString(0)
  Set strY = numY.ToString(0)
  '//
  '// 文字列を連結
  '//
  Set strPos = m.ConcatString(strX, m.DAString(", "))
  Set strPos = m.ConcatString(strPos, strY)
  '//
  '// 文字列からイメージを作成
  '//
  Set fnt = m.Font("Tahoma", 48, m.Blue).Bold
  Set img = m.StringImageAnim(strPos, fnt)
  '//
  '// コントロールにオブジェクトをセットしてスタート
  '//
  DAControl.Image = img
  DAControl.Start
  Call Window_onResize()
End Sub

'--------------------------------------------------------------
'ウィンドウのサイズ変更時の処理
'--------------------------------------------------------------
Sub Window_onResize()
  On Error Resume Next
  DAControl.Width  = Document.Body.ClientWidth
  DAControl.Height = Document.Body.ClientHeight
End Sub

</SCRIPT>
<! DirectAnimation コントロール>
<OBJECT ID = "DAControl"
  CLASSID = "CLSID:B6FFC24C-7E13-11D0-9B47-00C04FC2F51D"
  STYLE = "position:absolute; left:0; top:0; z-index:-1">
</OBJECT>
</BODY>
</HTML>
```

《リスト2》サンプル『マウスのクリック・イベント』"mosClick.htm"（VBScript用）

```
<HTML>
<TITLE>マウスのクリックイベント</TITLE>
<BODY LEFTMARGIN = "0", TOPMARGIN = "0">
<SCRIPT LANGUAGE = "VBScript">
'//--------------------------------------------------------------
'//
'// DirectAnimation Sample                        by Masato Hirai
'//
'//    『マウスのクリックイベント』
'//
'//--------------------------------------------------------------
Option Explicit

'//--------------------------------------------------------------
'// 初期化処理
'//--------------------------------------------------------------
Sub Window_onLoad()
  '//
  '// 変数・定数の宣言
  '//
  Dim m       'As DAStatics    'DAStaticsオブジェクト
  Dim pie     'As DAPath2      '扇形の軌跡
  Dim panel   'As DAImage      '塗りつぶし用のべた塗りイメージ
  Dim border  'As DALineStyle  '枠線のスタイル
  Dim col     'As DAColor      '塗りつぶしの色
  Dim rate    'As DANumber     '回転／停止を制御する変動値
  Dim img     'As DAImage      '扇形イメージ
  '//
  '// DirectAnimationの初期化（ピクセル座標系）
  '//
  Set m = DAControl.PixelLibrary
  '//
  '// DAColorオブジェクトのインスタンスを作成
  '//
  Set col = CreateObject("DirectAnimation.DAColor")
  '//
  '// マウスの右クリックで変化させる設定
  '//
  col.Init m.Until(m.Blue, m.RightButtonDown, _
          m.Until(m.Cyan, m.RightButtonDown, _
          m.Until(m.Lime, m.RightButtonDown, col)))
  '//
  '// 塗りつぶし用のべた塗りイメージを作成
  '//
  Set panel = m.SolidColorImage(col)
  '//
  '// 扇形イメージを作成
  '//
  Set border = m.DefaultLineStyle.Width(4)
  Set pie = m.PieDegrees(30, 330, 240, 240)
  Set img = pie.Fill(border, panel)
  '//
  '// イメージに回転(毎秒180度半時計回り)を与える
  '//
  Set img = img.Transform(m.Rotate2RateDegrees(180))
  '//
  '// DANumberオブジェクトのインスタンスを作成
  '//
  Set rate = CreateObject("DirectAnimation.DANumber")
  '//
  '// マウスの左クリックで[1]と[0]が切り替わる設定
  '//
  rate.Init m.Until(m.DANumber(1), m.LeftButtonDown, _
          m.Until(m.DANumber(0), m.LeftButtonDown, rate))
  '//
  '// 回転のスタート／ストップを制御
  '//
  Set img = img.SubstituteTime(m.Integral(rate))
  '//
  '// コントロールにオブジェクトをセットしてスタート
  '//
  DAControl.Image = img
  DAControl.Start
  Call Window_onResize()
End Sub

'--------------------------------------------------------------
'ウィンドウのサイズ変更時の処理
'--------------------------------------------------------------
Sub Window_onResize()
  On Error Resume Next
  DAControl.Width  = Document.Body.ClientWidth
  DAControl.Height = Document.Body.ClientHeight
End Sub

</SCRIPT>
<! DirectAnimation コントロール>
<OBJECT ID = "DAControl"
  CLASSID = "CLSID:B6FFC24C-7E13-11D0-9B47-00C04FC2F51D"
  STYLE = "position:absolute; left:0; top:0; z-index:-1">
</OBJECT>
</BODY>
</HTML>
```

《リスト3》サンプル『フライング・バード』"FlyBird.htm"（VBScript用）

```
<HTML>
<TITLE>フライング・バード</TITLE>
<BODY LEFTMARGIN = "0", TOPMARGIN = "0">
<SCRIPT LANGUAGE = "VBScript">
'//--------------------------------------------------------------
'//
'// DirectAnimation Sample                        by Masato Hirai
'//
'//    『フライング・バード』
'//
'//--------------------------------------------------------------
Option Explicit

'//--------------------------------------------------------------
'// 初期化処理
'//--------------------------------------------------------------
Sub Window_onLoad()
  '//
  '// 変数・定数の宣言
  '//
  Dim m       'As DAStatics   'DAStaticsオブジェクト
  Dim chra    'As DAImage     'キャラクタイメージ
  Dim mouse   'As DAPoint2    'マウスポインタの座標(x, y)
  Dim dx      'As DANumber    'X軸方向移動量
  Dim dy      'As DANumber    'Y軸方向移動量
  Dim birds   'As DAImage     'キャラクタの合成イメージ
  Dim sky     'As DAImage     '背景イメージ
  Dim eff     'As DASound     '効果音
  Dim bgm     'As DASound     'ＢＧＭ
  '//
  '// DirectAnimationの初期化（ピクセル座標系）
  '//
  Set m = DAControl.PixelLibrary
  '//
  '// 画像ファイル(アニメーションGIF)のロード
  '//
  Set chra = m.ImportImage("CH223.GIF")
  '//
  '// X軸方向(左右)とY軸方向(上下)の変動値を定義
  '//
  Set dx = m.Mul(m.Sin(m.LocalTime), m.DANumber(32))
  Set dy = m.Sin(m.Mul(m.LocalTime, m.DANumber(2.5)))
  Set dy = m.Mul(dy, m.DANumber(48))
  '//
  '// 移動を定義してキャラクタイメージに適用
  '//
  Set chra = chra.Transform(m.Translate2Anim(dx, dy))
  '//
  '// マウスポインタの座標を取得
  '//
  Set mouse = m.MousePosition
  '//
  '// マウスポインタの座標値を固定
  '//
  Set mouse = m.UntilEx(mouse, m.Always.Snapshot(mouse))
  '//
  '// キャラクタをマウスポインタの位置に移動
  '//
  Set chra = chra.Transform(m.Translate2Point(mouse))
  '//
  '// マウスのクリック時にイメージをつみ重ねる処理
  '//
  Set birds = CreateObject("DirectAnimation.DAImage")
  birds.Init m.Until(m.EmptyImage, m.LeftButtonDown, _
                  m.Overlay(birds, chra))
  '//
  '// 背景イメージ(斜めにスクロール)
  '//
  Set sky = m.ImportImage("BCG_024.GIF").Tile
  Set sky = sky.Transform(m.Translate2Rate(-50, 10))
  '//
  '// 効果音(クリックのたびに再生)
  '//
  Set eff = CreateObject("DirectAnimation.DASound")
  eff.Init m.Until(m.Silence, m.LeftButtonDown, _
          m.Until(m.ImportSound("bfire.wav").Sound, _
                  m.Timer(0.18), eff))
  '//
  '// ＢＧＭ(無限リピート再生)
  '//
  Set bgm = m.ImportSound("GM091.MID").Sound.Loop
  '//
  '// コントロールにオブジェクトをセットしてスタート
  '//
  DAControl.Image = m.Overlay(birds, sky)       '← イメージ
  DAControl.Sound = m.Mix(eff, bgm)             '← サウンド
  DAControl.Start
  Call Window_onResize()
End Sub

'--------------------------------------------------------------
'ウィンドウのサイズ変更時の処理
'--------------------------------------------------------------
Sub Window_onResize()
  On Error Resume Next
  DAControl.Width  = Document.Body.ClientWidth
  DAControl.Height = Document.Body.ClientHeight
End Sub

</SCRIPT>
<! DirectAnimation コントロール>
<OBJECT ID = "DAControl"
  CLASSID = "CLSID:B6FFC24C-7E13-11D0-9B47-00C04FC2F51D"
  STYLE = "position:absolute; left:0; top:0; z-index:-1">
</OBJECT>
</BODY>
</HTML>
```

リスト1～リスト3入力&実行方法

このリストは，IE5／5.5とテキスト・エディタ(『メモ帳』など）があれば，実行することが可能だ。IE5は，以下のアドレスから無償でダウンロードすることができる。

● http://www.microsoft.com/windows/ie_intl/ja/download/

リストをメモ帳などで入力したら，拡張子を"htm"または"html"にして保存する。そして，IE5の[ファイル]メニューより，[開く]で保存したファイルを読み込もう。すると，リストが実行されるようすを，IE5で確認することができる。

プログラマー&クリエイターへの登竜門

BM特選 投稿プログラムコーナー

- Visual Basic
- Directx for Visual Basic
- Visual C++
- JAVA
- iアプリ
- ワンダーウィッチ
- FX-890P
- MUSIC

本コーナーは，さまざまな言語に対応したプログラミング学習と創作をテーマに，読者からの投稿作品で成り立っています。これから，プログラミングを勉強しようというみなさんや，プロのプログラマー／クリエイターを目指そうというみなさんにとって役立つ情報が満載です。手近にあるパソコンに入力して，お楽しみください。

それぞれの記事にはプログラムを入力／実行するときの注意点やプログラムの内容，改造方法などが紹介されていますので，よく読んでください。また，“CHECKER FLAG” というコラムがあり，デバッグ博士ことDr.Dがそれぞれのプログラムに対し批評を加えています。口が悪い人なので，かなりキビシイことも言っていますが，自分でプログラムを作ったり，これから投稿しようとしている人にとっては，とても参考になるでしょう。



プログラム投稿者へのお願い

自分の作ったプログラムを投稿しようという人はつぎのきまりを守って応募してください。

①プログラム（作品）ファイル ②プログラムの説明，住所・氏名・年齢・職業（学校名）・電話番号（連絡のつく時間も）を書いた原稿のテキスト・ファイルをフロッピーディスクなどにセーブし（プリンタで打ち出した原稿も同封のこと），お送りください。

プログラムの説明には，①内容 ②使いかた・遊びかた ③操作方法 ④プログラムのしくみ（変数表やフローチャートも）⑤プログラムの具体的入力方法，注意点 ⑥改造のアドバイス，作ったとき苦労した点 ⑦使用機種名 ⑧言語（バージョン，モードなども）などを書きます。

注1）プログラムは必ずチェックし，ミスを防ぐために同じものを2～3入れてもらえば安心です。

注2）プログラムの説明は，テキスト・ファイルをフロッピーディスクなどにセーブし，念のためプリンタで打ち出したものも同封することが基本ですが，手書き原稿でも受け付けます。その際，できるだけていねいに，マス目のある用紙を使用してください。

注3）ディスクのラベル，封筒の裏面にもプログラム名（ファイル名も），原稿ファイル名，住所，氏名，電話番号を必ず記入してください。

内容はオリジナルのもので，未発表のものに限ります。もし，参考にした記事や本がありましたら，書名，号数，記事名，筆者名，記事のタイトルなども明記しておいてください。

お送りいただいたディスクはお返しできませんので，あらかじめバックアップをとっておいてください。

掲載されたプログラムに対しては，規定の原稿料をお支払いし，掲載誌をお送りいたします。　**編集部**

★本誌とじ込みの愛読者カードについている応募カードも忘れずに同封してください。

あて先：〒141-8715東京都品川区東五反田1-11-15　電波新聞社マイコンBASICマガジン「特選プログラム・コーナー」係

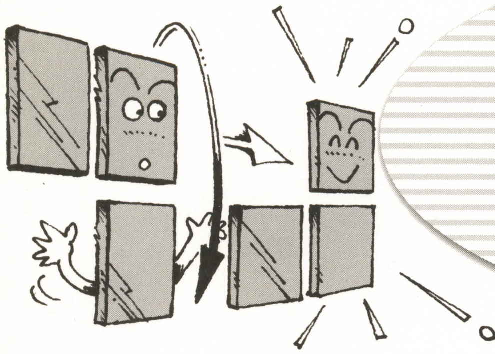
★ここに掲載されているプログラムは，ベーマガ・ホームページからもダウンロードできます。また，ホームページからの投稿も受け付けています。

http://www.basicmagazine.net/

Windows(Visual Basic V6.0)用

回転の動きを把握しよう

くるっと回転（かいてん）



小泉　賢一（14歳）

内容

16枚の色のついたパネルを操作して，お手本と同じような色の配置にパネルを移動させるパズルゲームです。レベルによって難しさが違います。ルービックキューブの２次元版みたいなものです。

ルール・操作説明

最初にオープニングでレベルを選びます（レベルによってパネルの色数および枚数が違います）。するとゲーム画面が現れます。スタートボタンを押すとはじまりです。

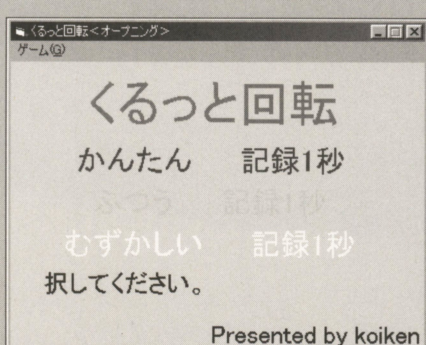
４×４の正方形に並んだ16枚のパネルが２組あり，左側がプレイヤーの操作するパネルで，右側がお手本です。

プレイヤー側のパネルには９個のボタンがあり，ボタンをクリックすると，そのまわりの四つのパネルが回転します。左クリックで左回転，右クリックで右回転です。また，お手本も10秒に１回，四つのパネルがランダムに回転します。何回か回転することによりお手本と同じパネルの配置にすればクリアです。

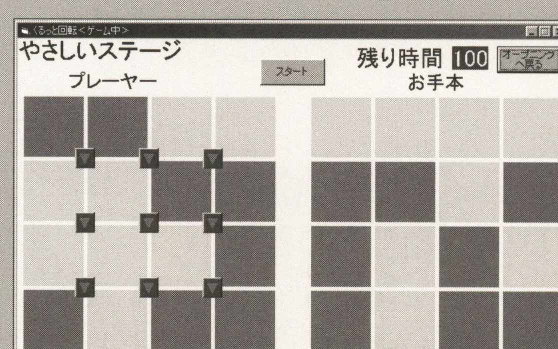
プログラムの構造と注意点

初回起動時にステージデータファイル（stagedata.txt）と，スコアファイル（score.txt）をプログラムと同じフォルダに作成します。また，あらかじめ（button.bmp）という25×25ピクセルのボタン用画像ファイルが必要です。とくに模様は必要なく真っ白でもＯＫです。

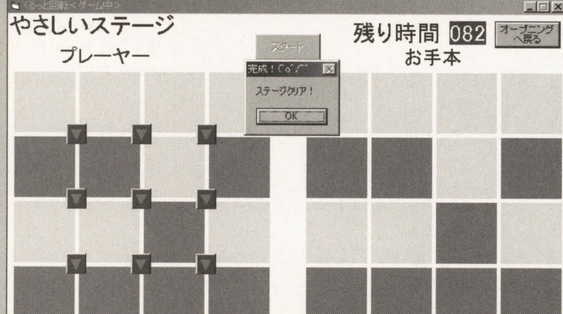
制限時間やアニメーションはタイマーコントロールを使っています。だいたい１秒間をInterval値で1000として設定している



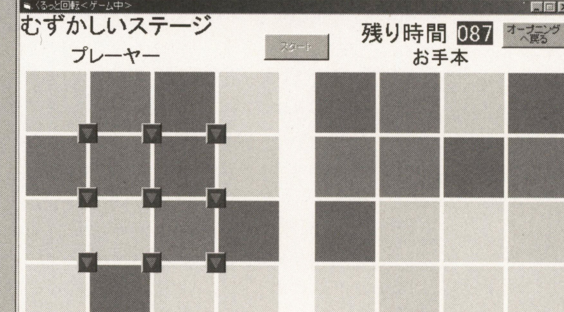
《写真１》三つのレベルが選択可能



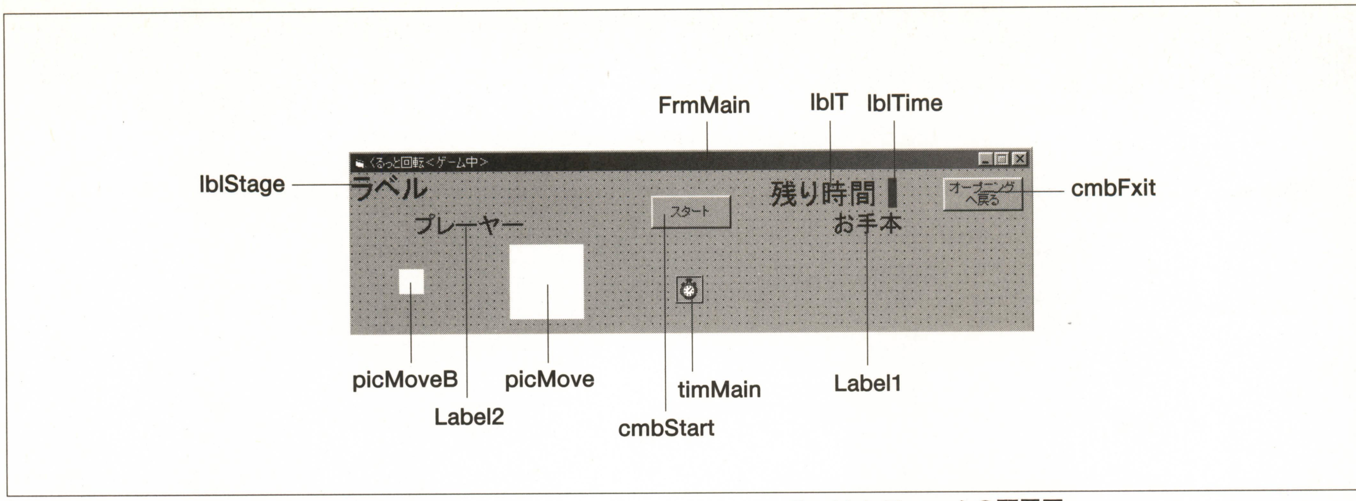
《写真２》右側と同じ配置になるようにする



《写真３》短時間クリアを目指そう



《写真４》レベルが上がるとパネルの色数が増える



FrmMainフォームの配置図

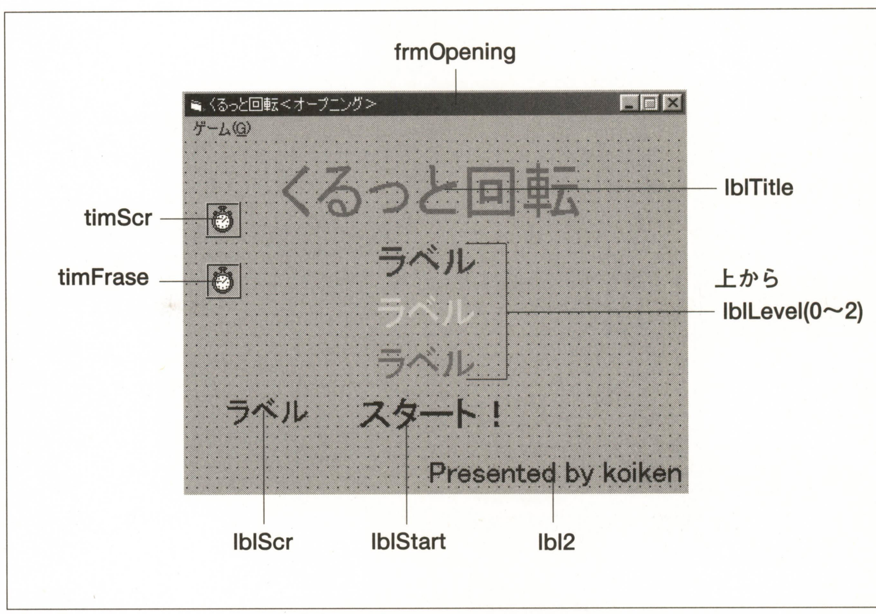

button.bmpの画像

ので，速さが合わないときは適当に変えてください。ちなみに，プログラム中でインデックスが0のコントロールはまったく使っていません。コントロールをロードするときに必要なだけです。

その他

改造点としては，同じような命令をなるべく簡単に済ませるようにする，パネルの回転するアニメーションをつけるなどがあると思います。

最後にプログラミングの世界に引き込んでくれた竹内様，VisualBasicを教えてくれた高橋様，ありがとうございました。



FrmOpeningフォームの配置図

《第１表》frmOpeningのプロパティ表

フォームの設定	
Name	frmOpening
AutoRedraw	-1 'True
Caption	くるっと回転<オープニング>
Height	4830
Left	0
LinkTopic	Form1
MaxButton	0 'False
ScaleHeight	276
ScaleMode	3 'ピクセル
ScaleWidth	398
StartUpPosition	2 '画面の中央
Top	0
Width	6090
ラベルの設定	
Name	lblStart
Alignment	2 '中央揃え
AutoSize	-1 'True
BackStyle	0 '透明
Caption	スタート！
Font	ＭＳ Ｐゴシック
Height	29
Left	137
TabIndex	6
Top	200
Width	125
Name	lblScr
AutoSize	-1 'True
BackStyle	0 '透明
Caption	ラベル
Font	ＭＳ Ｐゴシック
Height	24

Left	32
TabIndex	5
Top	200
Width	361
Name	lblTitle
Alignment	2 '中央揃え
AutoSize	-1 'True
BackStyle	0 '透明
Caption	くるっと回転
Font	ＭＳ Ｐゴシック
ForeColor	&H000000FF&
Height	48
Left	73
TabIndex	4
Top	16
Width	243
Name	lblLevel(0)
ForeColor	&H00FF0000&
Index	0
TabIndex	1
Top	80
Name	lblLevel(1)
ForeColor	&H0000FF00&
Index	1
TabIndex	2
Top	120
Name	lblLevel(2)
ForeColor	&H0000FFFF&
Index	2
TabIndex	3
Top	160
Name	lblLevel(0〜2)

Alignment	2 '中央揃え
AutoSize	-1 'True
BackStyle	0 '透明
Caption	ラベル
Font	ＭＳ Ｐゴシック
Height	29
Left	151
Width	81
Name	lbl2
Alignment	1 '右揃え
AutoSize	-1 'True
BackStyle	0 '透明
Caption	Presented by koiken
Font	ＭＳ Ｐゴシック
ForeColor	&H00000000&
Height	24
Left	192
TabIndex	0
Top	248
Width	202
タイマーの設定	
Name	timErase
Enabled	0 'False
Interval	1
Left	16
Top	96
Name	timScr
Enabled	0 'False
Interval	80
Left	16
Top	48

★うつくしま未来博と山口きらら博，編集部のみなさんは，どの博覧会がイチオシですか？（福島県郡山市・**近内沖尋18歳**）……【**編集長**：それぞれ会期も9月30日までだし，一度見ておくとしようか。私と編君は福島へ，Dr.とつぐ美くんは山口へ。**影**：ちょっとー。】

《第2表》frmMainのプロパティ表

フォームの設定	
Name	frmMain
AutoRedraw	-1 'True
Caption	くるっと回転<ゲーム中>
Height	2805
Left	0
LinkTopic	Form1
MaxButton	0 'False
ScaleHeight	160
ScaleMode	3 'ピクセル
ScaleWidth	688
StartUpPosition	2 '画面の中央
Top	0
Width	10440
ピクチャボックスの設定	
Name	picMove
Appearance	0 'フラット
AutoRedraw	-1 'True
BackColor	&H80000005&
BorderStyle	0 'なし
ForeColor	&H80000008&
Height	75
Index	0
Left	160
ScaleHeight	75
ScaleMode	3 'ピクセル
ScaleWidth	75
TabIndex	7
Top	72
Width	75
Name	picMoveB
Appearance	0 'フラット
BackColor	&H80000005&
BorderStyle	0 'なし
ForeColor	&H80000008&
Height	25
Index	0
Left	48
ScaleHeight	25
ScaleMode	3 'ピクセル
ScaleWidth	25
TabIndex	8
Top	96
Width	25
コマンドボタンの設定	
Name	cmbStart
Caption	スタート
Height	33
Left	304
TabIndex	6
Top	24
Width	81
Name	cmbExit
Caption	オープニングへ戻る
Height	33
Left	600
TabIndex	5
Top	8
Width	81
ラベルの設定	
Name	lblStage
AutoSize	-1 'True
BackStyle	0 '透明
Caption	ラベル
Font	ＭＳ Ｐゴシック
ForeColor	&H00000000&
Height	29
Left	0
TabIndex	4
Top	0
Width	79
Name	lblTime
AutoSize	-1 'True
BackColor	&H00FF0000&
Font	ＭＳ Ｐゴシック
ForeColor	&H0000FFFF&
Height	29
Left	544
TabIndex	3
Top	8
Width	9
Name	lblT
AutoSize	-1 'True
BackStyle	0 '透明
Caption	残り時間
Font	ＭＳ Ｐゴシック
Height	29
Left	424
TabIndex	2
Top	8
Width	109
Name	Label1
AutoSize	-1 'True
BackStyle	0 '透明
Caption	お手本
Font	ＭＳ Ｐゴシック
Height	24
Left	488
TabIndex	0
Top	40
Width	71
Name	Label2
AutoSize	-1 'True
BackStyle	0 '透明
Caption	プレーヤー
Font	ＭＳ Ｐゴシック
Height	24
Left	64
TabIndex	1
Top	40
Width	111
タイマーの設定	
Name	timMain
Enabled	0 'False
Interval	1000
Left	328
Top	104

《第3表》メニュー(frmOpening)の設定

ｷｬﾌﾟｼｮﾝ(P)	名前(M)	ｲﾝﾃﾞｯｸｽ(X)	ﾍﾙﾌﾟｺﾝﾃｷｽﾄ番号(H)	ｼｮｰﾄｶｯﾄ(S)	ﾈｺﾞｼｴｰﾄ位置(O)	ﾁｪｯｸ(C)	有効 (E)	表示(V)	ｳｲﾝﾄﾞｳﾘｽﾄ(W)
ゲーム(&G)	mnuGame		0	(なし)	0ーなし		○	○	
…かんたん(&E)	mnuEasy		0	F2	0ーなし		○	○	
…ふつう(&N)	mnuNormal		0	F3	0ーなし		○	○	
…むずかしい(&H)	mnuHard		0	F4	0ーなし		○	○	
…終了(&Q)	mnuExit		0	(なし)	0ーなし		○	○	

CHECKER FLAG

影：最近はこういった，回転の動きを取り入れたパズルの作品が目に付くなぁ～。ただ，こういう回転を取り入れるのであれば，その動きをアニメーションで再現してほしい。パネルの場所が変わるだけだと，回転したのかどうかが，わかりにくいからね。

つぐ美：ゲームとしては，お手本のパネルも変化していくので，クリアまでは運に左右されやすいわね。より多くの動きを求めて，この機能を入れたのかもしれないけど，お手本パネルは変化しないほうがいいと思うわ。

Dr.D：コントロールをリストからLoadしているので，フォームはコンパクトにまとまっておるが，ゲーム内容の割にはリストが長めじゃな～。改善点としては，説明文にもあるように同じような命令をうまくまとめることじゃな。たとえば，paneKaitenプロシージャで言えば，Select Case文は使わず，ButtonNumberの値を直接引数に使用するようにすれば，Call CHGの記述量が減ってくるじゃろう。

《リスト1》frmMain.frmのプログラム・リスト

```
Option Explicit
Dim i As Integer                        'ＦＯＲ文用
Dim j As Integer                        'ＦＯＲ文用
Dim MOV(33) As Integer                  'プレーヤー側パネルの種類の値（０赤、１水、２紫、３緑)
Dim paneMax(3) As Integer               '各色の最大パネル数（パネル数の上限)
Dim dpane(3) As Integer                 '設定中の使用パネル数（現在の決定パネル数)'
Dim pCount As Integer                   'パネルの色が決定したかしてないか
Dim intRet As Integer                   '二つの変数の値入れ替えに使う
Dim readPane As String                  'ファイルから読み込んだ最大パネル数
Dim restTime  As Integer                '残り時間
Dim chgPaneTime As Integer              'お手本パネル移動の時間の間隔
Dim oldTime(3) As Integer               '更新前のスコア
Const paneKindMax = 4                   'パネルの種類（使用しないものも含む)
Dim paneKind As Integer                 'そのステージのパネルの種類（使用するものだけ)
Dim msgBack                             'メッセージボックスの戻り値

'    @@@@@「オープニングへ戻る」ボタンクリック時の処理@@@@@
Private Sub cmbExit_Click()
    Unload Me
    frmOpening.Show
End Sub

'    @@@@@スタートボタンクリック時の処理@@@@@
Private Sub cmbStart_Click()
    cmbStart.Enabled = False
    For i = 1 To (paneKindMax - 1) ^ 2
        picMoveB(i).Enabled = True
    Next i
'    @@@@@制限時間カウント開始@@@@@
    timMain.Enabled = True
End Sub

Private Sub Form_Load()
'    @@@@@コントロールのロード@@@@@
    For i = 1 To 9
        Load picMoveB(i)
        picMoveB(i).Left = 72 + 80 * ((i - 1) Mod 3)
        picMoveB(i).Top = 136 + 80 * (((i - 1) Mod 9) ¥ 3)
        picMoveB(i).Width = 25
        picMoveB(i).Height = 25
        picMoveB(i).Visible = True
    Next i
    For i = 1 To 32
        Load picMove(i)
```


★どうやら私は「地獄の雷（いかづち）」を呼び寄せてしまったらしい。（愛知県一宮市・幻18歳）……【**影**：ぐわーっ！ヒットポイントが極端になくなったため，仕事しばし休憩します。**編集長**：編くん，ベホマを影くんに。**編**：回復の魔法ですね。**影**：遠慮しとくよ。】

```
        picMove(i).Left = 8 + 80 * ((i - 1) Mod 4) + ((i - 1) ¥ 16) * 360
        picMove(i).Top = 72 + 80 * (((i - 1) Mod 16) ¥ 4)
        picMove(i).Width = 75
        picMove(i).Height = 75
        picMove(i).Visible = True
    Next i
'     @@@@@変数設定@@@@@
    Me.Width = 10440
    Me.Height = 6315
    picMove(0).Visible = False
    picMoveB(0).Visible = False
    Randomize Val(Right(Time$, 2))
    Me.BackColor = vbYellow
    Open frmOpening.dirpath & "stagedata.txt" For Input As #5
    For i = 1 To frmOpening.stage
        Input #5, readPane
    Next i
    Close #5
    For i = 0 To 3
        paneMax(i) = Val(Right(Left(readPane, i + 1), 1))
    Next i
    restTime = 100
    chgPaneTime = 10
    paneKind = 4
    lblTime.Caption = Right("000" & Right(Str(restTime), Len(Str(restTime)) - 1), 3)
    Select Case frmOpening.stage
        Case 1
            lblStage.Caption = "やさしいステージ"
        Case 2
            lblStage.Caption = "ふつうステージ"
        Case 3
            lblStage.Caption = "むずかしいステージ"
    End Select
'     @@@@@プレイヤー側のパネルの初期配置設定@@@@@
    For i = 0 To 3
        dpane(i) = 0
    Next i
    i = 1
    Do While (i < (paneKindMax ^ 2 + 1))
        DoEvents
        MOV(i) = Int(Rnd * paneKind)
        pCount = 0
        For j = 0 To 3
            If MOV(i) = j And dpane(j) < paneMax(j) And pCount = 0 Then MOV(i) = j: i = i + 1:
dpane(j) = dpane(j) + 1: pCount = 1
        Next j
    Loop
    For i = 1 To paneKindMax ^ 2
        Call PicLoad(i)
    Next i
    For i = 1 To (paneKindMax - 1) ^ 2
        picMoveB(i).Picture = LoadPicture(frmOpening.dirpath & "button.bmp")
        picMoveB(i).Enabled = False
    Next i
'     @@@@@お手本側のパネル配置設定@@@@@
    For i = 0 To 3
        dpane(i) = 0
    Next i
    i = 1
    Do While (i < (paneKindMax ^ 2 + 1))
        DoEvents
        MOV(i + 16) = Int(Rnd * paneKind)
        pCount = 0
        For j = 0 To 3
            If MOV(i + 16) = j And dpane(j) < paneMax(j) And pCount = 0 Then MOV(i + 16) = j: i
= i + 1: dpane(j) = dpane(j) + 1: pCount = 1
        Next j
    Loop
    For i = 1 + 16 To paneKindMax ^ 2 + 16
        Call PicLoad(i)
    Next i
End Sub
```

```
'     @@@@@回転ボタンクリック時の処理@@@@@
Private Sub picMoveB_MouseDown(Index As Integer, Button As Integer, Shift As Integer, X As
Single, Y As Single)
    Dim chg(4) As Integer
    picMoveB(Index).Enabled = False
'     @@@@@押されたボタンによる変数と表示の回転処理@@@@@
    Call paneKaiten(Index, Button, 0)
    picMoveB(Index).Enabled = True
'     @@@@@クリア確認処理@@@@@
    For i = 1 To 16
        If MOV(i) <> MOV(i + 16) Then Exit For
        If i = 16 Then Call sCLEAR
    Next i
End Sub

'     @@@@@パネルを回転させる処理@@@@@
Sub sCHG(pp1 As Integer, pp2 As Integer, pp3 As Integer, pp4 As Integer, sButton As Integer,
plus As Integer)
'     @@@@@「pp1,pp2,pp3,pp4」はくるっと回転させるパネルのインデックス番号@@@@@
'     @@@@@「sButton」は左回転か右回転か@@@@@
'     @@@@@「plus」はプレイヤー側なら「0」、お手本側なら「1」@@@@@
    Dim chg(4) As Integer
            chg(1) = pp1 + plus * 16
            chg(2) = pp2 + plus * 16
            chg(3) = pp3 + plus * 16
            chg(4) = pp4 + plus * 16
            If sButton = 1 Then '     @@@@@左回転なら@@@@@
                intRet = MOV(chg(1))
                MOV(chg(1)) = MOV(chg(2))
                MOV(chg(2)) = MOV(chg(3))
                MOV(chg(3)) = MOV(chg(4))
                MOV(chg(4)) = intRet
                For i = 1 To 4
                    Call PicLoad(chg(i))
                Next i
            End If
            If sButton = 2 Then '     @@@@@右回転なら@@@@@
                intRet = MOV(chg(1))
                MOV(chg(1)) = MOV(chg(4))
                MOV(chg(4)) = MOV(chg(3))
                MOV(chg(3)) = MOV(chg(2))
                MOV(chg(2)) = intRet
                Call PicLoad(chg(1))
                Call PicLoad(chg(4))
                Call PicLoad(chg(3))
                Call PicLoad(chg(2))
            End If
End Sub

'     @@@@@クリア時の処理@@@@@
Sub sCLEAR()
    On Error Resume Next
    timMain.Enabled = False
    For j = 1 To 16
        picMove(j).Enabled = False
        picMove(j + 16).Enabled = False
    Next j
    For j = 1 To 9
        picMoveB(j).Enabled = False
    Next j
    msgBack = MsgBox("ステージクリア！", vbOKOnly, "完成！(^o^/~~")
'     @@@@@スコア更新かどうかの確認@@@@@
    Open frmOpening.dirpath & "score.txt" For Input As #6
    For j = 1 To 3
        Input #6, oldTime(j)
    Next j
    Close #6
    If oldTime(frmOpening.stage) < restTime Then
        oldTime(frmOpening.stage) = restTime
        Open frmOpening.dirpath & "score.txt" For Output As #6
        For j = 1 To 3
```

★少しプログラミングをしてみたいなと思っていました。これならできるかも…。（山口県玖珂郡・**瀬六40歳**）……**【編**：プログラムに興味を持ってもらえて，非常にうれしいです。こうした初心者に向けた記事も，どんどん増やしていきますので，お楽しみに！**】**

```
            Print #6, oldTime(j)
        Next j
        Close #6
        msgBack = MsgBox("スコア更新！！", vbOKOnly, "最短記録登場")
    End If
    frmOpening.Show
    Unload Me
End Sub

'     @@@@@制限時間関係の処理@@@@@
Private Sub timMain_Timer()
    restTime = restTime - 1
    lblTime.Caption = Right("000" & Right(Str(restTime), Len(Str(restTime)) - 1), 3)
    If restTime = -1 Then
'     @@@@@時間切れの時の処理@@@@@
        lblTime.Caption = "000"
        For i = 1 To 16
            picMove(i).Enabled = False
            picMove(i + 16).Enabled = False
        Next i
        For i = 1 To 9
            picMoveB(i).Enabled = False
        Next i
        msgBack = MsgBox("時間切れです。", vbOKOnly, "ゲームオーバー")
        Unload Me
        frmOpening.Show
    End If
'     @@@@@お手本パネルの定時変化による変数と表示の回転処理@@@@@
    If Int(restTime / chgPaneTime) * chgPaneTime = restTime Then Call paneKaiten(Int(Rnd * 9) +
1, 0, 1)
End Sub

'     @@@@@パネルの色変更関数@@@@@
Sub PicLoad(paneN As Integer)
'     @@@@@「paneN」は色変更するパネルのインデックス番号@@@@@
    Select Case MOV(paneN)
        Case 0
            picMove(paneN).BackColor = vbRed
        Case 1
            picMove(paneN).BackColor = vbCyan
        Case 2
            picMove(paneN).BackColor = vbMagenta
        Case 3
            picMove(paneN).BackColor = vbGreen
    End Select
End Sub

Sub paneKaiten(ButtonNumber As Integer, sButton As Integer, plus As Integer)
'     @@@@@「ButtonNumber」は押されたボタンのインデックス番号（お手本なら仮想ボタンのインデ
ックス番号）@@@@@
'     @@@@@「sButton」は「0」でランダム「1」で左回転、「2」で右回転@@@@@
'     @@@@@「plus」はプレイヤー側なら「0」、お手本側なら「1」@@@@@
    If sButton = 0 Then sButton = Int(Rnd * 2) + 1
    Select Case ButtonNumber
        Case 1
            Call sCHG(1, 2, 6, 5, sButton, plus)
        Case 2
            Call sCHG(2, 3, 7, 6, sButton, plus)
        Case 3
            Call sCHG(3, 4, 8, 7, sButton, plus)
        Case 4
            Call sCHG(5, 6, 10, 9, sButton, plus)
        Case 5
            Call sCHG(6, 7, 11, 10, sButton, plus)
        Case 6
            Call sCHG(7, 8, 12, 11, sButton, plus)
        Case 7
            Call sCHG(9, 10, 14, 13, sButton, plus)
        Case 8
            Call sCHG(10, 11, 15, 14, sButton, plus)
        Case 9
            Call sCHG(11, 12, 16, 15, sButton, plus)
```

```
    End Select
End Sub
```

《リスト２》frmOpening.frmのプログラム・リスト

```
Option Explicit
Const scrMozi = "                              レベルを選択してください。
"
'@@@@@スクロール文字@@@@@
Const space = 15
'@@@@@スクロール文の端のスペースの数@@@@@
Public dirpath As String            'プログラムのあるフォルダのパス
Public stage As Integer             'ステージ数（０かんたん、１ふつう、２むずかしい）
Dim i As Long                       'ＦＯＲ文用
Dim sFlag As Integer                'スクロール文字のどこの部分を表示するか
Dim cFlag As Integer                '選択肢の消え行くときのＲＧＢ変化のメーター
Dim msgBack                         'メッセージボックスの戻り値
Dim score(3) As Integer             'スコア表示

Private Sub Form_Load()
'     @@@@@プログラムのあるフォルダのパス取得@@@@@
    dirpath = App.Path
    If Right(dirpath, 1) <> "¥" Then dirpath = dirpath & "¥"
'     @@@@@データファイル検索＆作成@@@@@
    If Len(Dir(dirpath & "stagedata.txt")) <= 1 Then
        msgBack = MsgBox("ステージデータファイルが見つかりません。" & vbCrLf & "ステージデータ
ファイルを作成します。", vbYesNo, "初回起動時は必要なデータファイルを作成します。")
    End If
    If msgBack = 6 Then
        Open dirpath & "stagedata.txt" For Output As #2
        Print #2, 8800
        Print #2, 8440
        Print #2, 4444
        Close #2
    End If
    If msgBack = 7 Then
        msgBack = MsgBox("データがないと起動できません。" & vbCrLf & "終了します。", vbOKOnly,
"終了")
        End
    End If
    If Len(Dir(dirpath & "score.txt")) <= 1 Then
        msgBack = MsgBox("スコアファイルが見つかりません。" & vbCrLf & "スコアファイルを作成し
ます。", vbYesNo, "初回起動時は必要なデータファイルを作成します。")
    End If
    If msgBack = 6 Then
        Open dirpath & "score.txt" For Output As #2
        For i = 1 To 3
            Print #2, 1
        Next i
        Close #2
        msgBack = MsgBox("プログラムの再起動をします。", vbOKOnly, "再起動")
        Shell (dirpath & "kaiten.exe")
        End
    End If
    If msgBack = 7 Then
        msgBack = MsgBox("データがないと起動できません。" & vbCrLf & "終了します。", vbOKOnly,
"終了")
        End
    End If
'     @@@@@ハイスコア読みこみ@@@@@
    Open dirpath & "score.txt" For Input As #1
    For i = 1 To 3
        Input #1, score(i)
    Next i
    Close #1
'     @@@@@変数初期設定@@@@@
    Me.BackColor = RGB(0, 255, 255)
    lblLevel(0).Caption = "かんたん    記録" & score(1) & "秒"
    lblLevel(1).Caption = "ふつう     記録" & score(2) & "秒"
    lblLevel(2).Caption = "むずかしい    記録" & score(3) & "秒"
```

★ここで一句（8月号の続き）。“カゲさんの　貧乏さで　貧乏神も　近づけない　なぜだろうね”お粗末でした。（新潟県三島郡・フリーゲーマー13歳）……**【影：うーん，5·7·5·7·7の句をまったく無視しているところが，何とも言えない。編：日ごろの影さんのよう。】**

```
    lblStart.Visible = False
    sFlag = 0
    cFlag = 0
'     @@@@@説明文スクロール開始@@@@@
    timScr.Enabled = True
End Sub

Private Sub lbllevel_Click(Index As Integer)
'     @@@@@ボタンクリック@@@@@
    stage = Index + 1
    timScr.Enabled = False
    lblStart.Visible = True
    timErase.Enabled = True
End Sub

Private Sub mnuEasy_Click()
'     @@@@@メニューかんたんクリック@@@@@
    stage = 1
    timScr.Enabled = False
    lblStart.Visible = True
    timErase.Enabled = True
End Sub

Private Sub mnuExit_Click()
    End
End Sub

Private Sub mnuHard_Click()
'     @@@@@メニューむずかしいクリック@@@@@
    stage = 3
    timScr.Enabled = False
    lblStart.Visible = True
    timErase.Enabled = True
End Sub

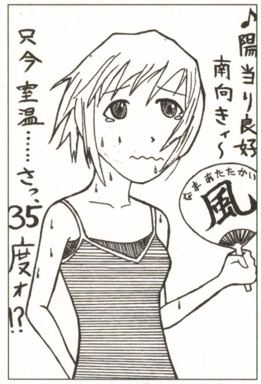
Private Sub mnuNormal_Click()
'     @@@@@メニューふつうクリック@@@@@
    stage = 2
    timScr.Enabled = False
    lblStart.Visible = True
    timErase.Enabled = True

End Sub

Private Sub timErase_Timer()
'     @@@@@ボタンクリック時の処理@@@@@
    On Error Resume Next
'     @@@@@選択しなかったラベルを消滅させる@@@@@
    Select Case stage
        Case 1
            lblLevel(0).ForeColor = RGB(0, 0, 255 - cFlag)
            lblLevel(1).ForeColor = RGB(0, 255, cFlag)
            lblLevel(2).ForeColor = RGB(255 - cFlag, 255, cFlag)
        Case 2
            lblLevel(0).ForeColor = RGB(0, cFlag, 255)
            lblLevel(1).ForeColor = RGB(0, 255 - cFlag, 0)
            lblLevel(2).ForeColor = RGB(255 - cFlag, 255, cFlag)
        Case 3
            lblLevel(0).ForeColor = RGB(0, cFlag, 255)
            lblLevel(1).ForeColor = RGB(0, 255, cFlag)
            lblLevel(2).ForeColor = RGB(255 - cFlag, 255 - cFlag, 0)
    End Select
    lblScr.ForeColor = RGB(0, cFlag, cFlag)
    lblStart.ForeColor = RGB(0, 255 - cFlag, 255 - cFlag)
    cFlag = cFlag + 14
    If cFlag >= 256 Then
        timErase.Enabled = False
        If stage <> 1 Then
            lblLevel(0).ForeColor = RGB(0, 255, 255)
            lblLevel(0).Visible = False
        End If
        If stage <> 2 Then
            lblLevel(1).ForeColor = RGB(0, 255, 255)
            lblLevel(1).Visible = False
        End If
        If stage <> 3 Then
            lblLevel(2).ForeColor = RGB(0, 255, 255)
            lblLevel(2).Visible = False
        End If
        lblScr.ForeColor = RGB(0, 255, 255)
        lblScr.Visible = False
        Unload Me
        frmMain.Show
    End If
End Sub

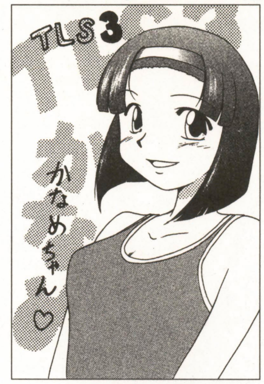
Private Sub timScr_Timer()
'     @@@@@説明文スクロール処理@@@@@
    sFlag = (sFlag + 1) Mod (Len(scrMozi) - space)
    lblScr.Caption = Left(Right(scrMozi, Len(scrMozi) - sFlag), space)
End Sub
```

評…ほんと，今年の夏は暑かったね（編）



by バグニンゲン

評…私は工藤翼子ファンかな。うっ，しまった（影）



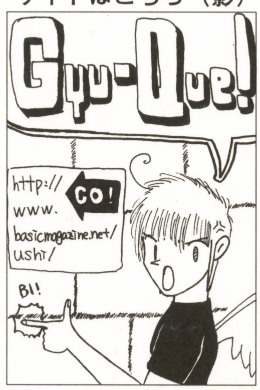
by ぶんちょ

評…伝言付きのイラスト投稿でした（編）



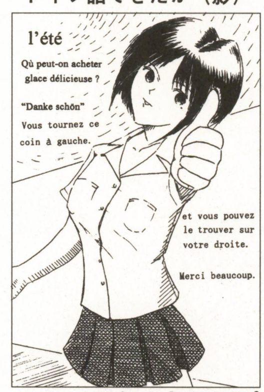
by 神奈月

評…人気の「牛食え」サイトはこちら（影）



by 挙動不審

評…私がわからんと思ってドイツ語できたか（影）



by みなみとらいぱ

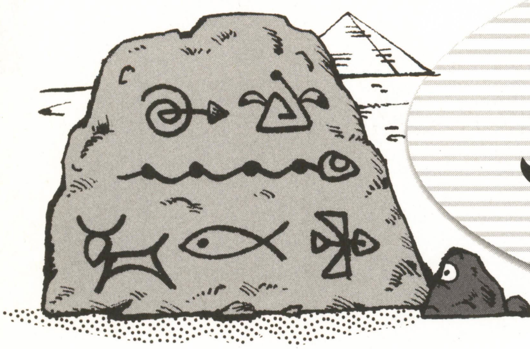
評…機会があったらまたよろしくね！（編）



by 鈴木真理子

★いつも楽しく読ませてもらっています。最近Excel VBAのプログラミング記事がたくさん載っているので，興味を持って見ています。VBAでリアルタイムゲームができたらすごいと思うのですが無理ですね。（島根県松江市・NSR250R+横浜ナンバー37歳）……【影：は。】

Windows(Visual Basic V6.0)用



石の移動先が重要

エジプトの石（いし）

峠　恒司（46歳）

はじめに

同じ石をペアで消していき，すべて消すとクリアという簡単なルールですが，すばやくやらないといけません。

ルール

・同じ石をペアで消していきます。すべての石が消えるとクリアです。
・石は動き出すと，他の石に当たるまで止まりません。同じ石に当たると，二つとも消えます。
・石が画面から飛び出すとミスです。
・最初は15秒以内でクリアしなければなりません。時間内にクリアすると石が二つずつ増えていきます。制限時間も長くなります。

マウス操作方法

動かしたい石をマウス左を押したままにします（ドラッグ）。マウスポインタは十字矢印になります。

そして，そのままマウスを上下左右に動かすと石は動かした方向に飛んでいきます（同じ石が隣あっているときは，すぐ消えます）。

マウスを右クリックすると，一手前に戻ります。何手前でも戻ります。

テクニック

消せる位置にあってもむやみに消してはだめです。消せる位置まですべての石を動かし，最後にすべて消すというのがベストの方法です。

プログラムについて

フォームにピクチャボックスを二つ貼り付けます。Picture2には石などの絵のビットマップを貼り付けます。縦横32×32ドットの絵を連続して適当に描きます。Picture1に石などをbitblt関数を使って表示します。

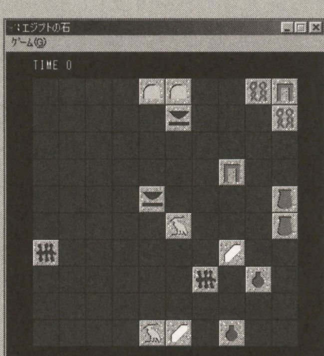
石の動くときの時間稼ぎ（ウェート）にtimeGetTime関数を使っています。このAPIは重宝しています。

効果音のWAVファイルは，いくつかありますが，適当に作ってください。

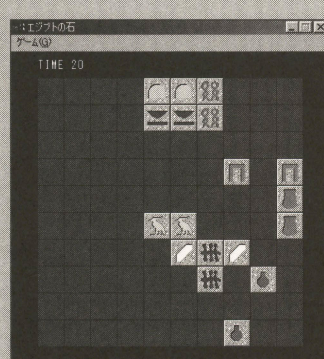
打ち込みやすいようにリストを短くするため，極力無駄をはぶいています。そのためタイトル画面などの飾りはありませんが，クリア後にエジプトといえば…あれが出ます。

その他

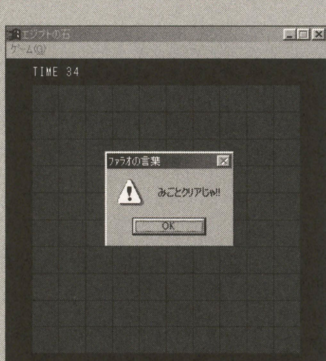
“ksu=9”のKSUを増やせば石の数が増えます。理論的には，50にするとすきまなく配置できますが，せいぜい40まで（石の数は80個）にしてください。



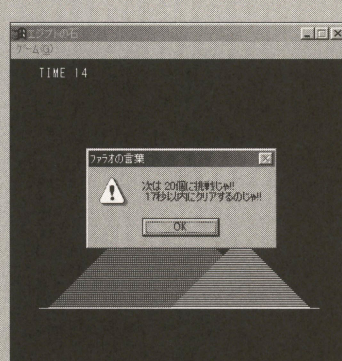
《写真1》同じ絵の石をぶつけていく



《写真2》ペアを作ってから消すのがコツ



《写真3》すべての石を消せばクリア



《写真4》指定タイム内クリアでつぎのステージへ

★1日3杯カキ氷。2kg増えても，3kg減。大丈夫？体重？！（愛知県名古屋市・**最馬場可？歳**）……【**編**：総入れ歯…，失礼，そう言えば，今年の夏はカキ氷食べなかったなー。**つぐ美**：私なんて，甘味やさん見つけるとすぐ，小倉のカキ氷よ。**編**：体重，大丈夫？】

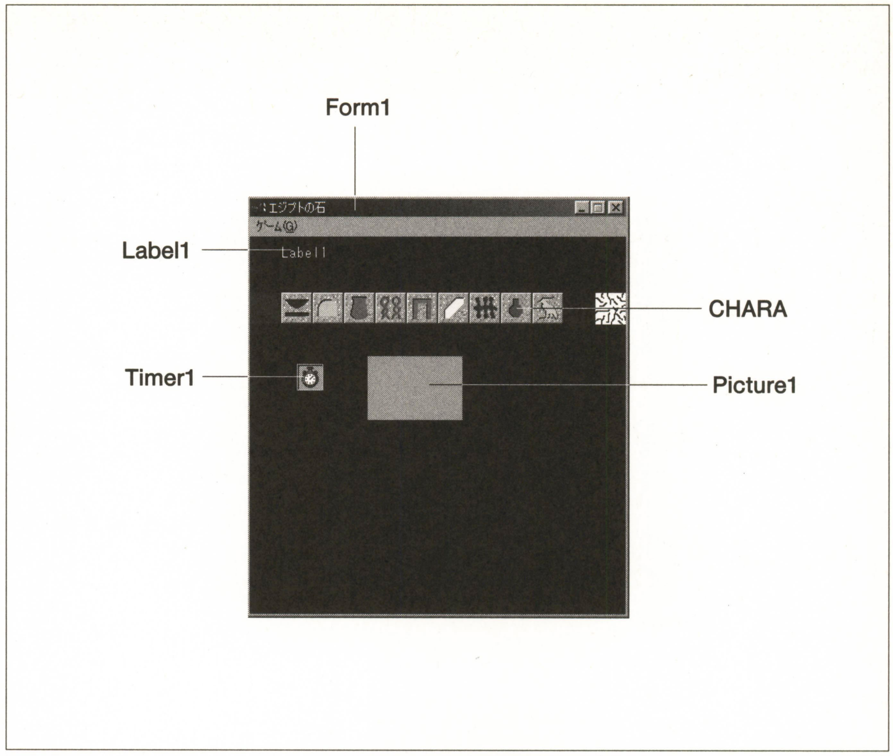
CHECKER FLAG

つぐ美：このタイトルに，見覚えのある読者もいるんじゃないかしら？そう，この作品は6月号で残念ながらボツになってしまった作品。今回は，そのときのProgrammer's Roomでのコメントにそった改良を行って，見事掲載となったわけです。

Dr.D：その前回からの修正点である，クリアできる配置の判断はNEWGAMEプロシージャ内のDo～Loop文のあたりで行っておるゾ。ループ関連の処理を多く使ってはおるが，比較的短い記述で改良を行っておる。コメントがないので多少わかりにくいかもしれんが，よく考えた処理と言える。

影：クリアタイムの基準が厳しいなぁ～。パズルを解く速度だけでなく，操作の素早さも要求されてしまうのはチョットいただけない。

編：その基準タイムの修正は，“If TMR <= 2 * KSU - 3 Then”の右側の式を書き換えればOKです。



フォームの配置図

《第1表》メニューの設定

ｷｬﾌﾟｼｮﾝ(P)	名前(M)	ｲﾝﾃﾞｯｸｽ(X)	ﾍﾙﾌﾟｺﾝﾃｷｽﾄ番号(H)	ｼｮｰﾄｶｯﾄ(S)	ﾈｺﾞｼｴｰﾄ位置(O)	ﾁｪｯｸ(C)	有効 (E)	表示(V)	ｳｲﾝﾄﾞｳﾘｽﾄ(W)
ｹﾞｰﾑ(&G)	GAME		0	(なし)	0 ーなし		○	○	
…ﾘﾌﾟﾚｲ(&R)	REPLAY		0	(なし)	0 ーなし		○	○	
…ﾆｭｰｹﾞｰﾑ(&N)	NEWG		0	(なし)	0 ーなし		○	○	
…-	BAR		0	(なし)	0 ーなし		○	○	
…ｴﾝﾄﾞ(&X)	ENDD		0	(なし)	0 ーなし		○	○	

おわりに

以前投稿したときのは，必ずクリアできるとは限りませんでした。今回のは必ずクリアできるように改良しました。しかし，簡単にクリアできるので，時間制限を設けました。タイトルはエジプトの石ですが，ほとんどゲームの内容とは関係ありません。



by ルストフレム

評…うまそうなスイカ！それにしてもデカイ（影）

《第2表》プロパティ表

フォームの設定	
Name	Form1
Appearance	0 'フラット
BackColor	&H00000000&
Caption	エジプトの石
ClipControls	0 'False
Font	ＭＳ ゴシック
ForeColor	&H00000000&
Height	6390
Icon	(アイコン)
Left	1440
MaxButton	0 'False
PaletteMode	1 'Z オーダー
ScaleHeight	380
ScaleMode	3 'ピクセル
ScaleWidth	383
Top	1635
Width	5865

ピクチャボックスの設定	
Name	Picture1
Appearance	0 'フラット
AutoRedraw	-1 'True
AutoSize	-1 'True
BackColor	&H00C0C0C0&
BorderStyle	0 'なし
Font	ＭＳ Ｐゴシック
ForeColor	&H80000008&
Height	65
Left	120
ScaleHeight	65
ScaleMode	3 'ピクセル
ScaleWidth	97
TabIndex	1
Top	120
Width	97
Name	CHARA
Appearance	0 'フラット
AutoRedraw	-1 'True
AutoSize	-1 'True
BackColor	&H80000005&
BorderStyle	0 'なし
ClipControls	0 'False
Enabled	0 'False
Font	ＭＳ Ｐゴシック
ForeColor	&H80000008&
Height	32
Left	0
Picture	(ビットマップ)
ScaleHeight	32
ScaleMode	3 'ピクセル
ScaleWidth	384
TabIndex	0
Top	56
Visible	0 'False
Width	384

ラベルの設定	
Name	Label1
AutoSize	-1 'True
BackStyle	0 '透明
Caption	Label1
Font	ＭＳ ゴシック
ForeColor	&H00FFFFFF&
Height	15
Left	32
TabIndex	2
Top	8
Width	48

タイマーの設定	
Name	Timer1
Left	48
Top	128

★人前で，指コンピュータで“132”を表すと喧嘩になるらしい！？（愛媛県東宇和郡・毛利祐允17歳）……【影：ベーマガにハガキを出そうとすると，最低1回は修正液の世話になるとのコメントもありましたが，このハガキは，3回使ってました。念のため。】

《リスト1》フォームのプログラム・リスト

```
Private Sub ENDD_Click()

End

End Sub

Private Sub Form_Load()

Form1.Left = (Screen.Width - Form1.Width) ¥ 2
Form1.Top = (Screen.Height - Form1.Height) ¥ 2
Picture1.Left = 32: Picture1.Top = 32
Picture1.Width = 320: Picture1.Height = 320
Form1.BackColor = &H0: Picture1.BackColor = &H0
KSU = 9
Call NEWGAME

End Sub


Private Sub MAKE()

'石を置く
Picture1.Cls
For M = 1 To 10: For N = 1 To 10: K = C(M, N): Call PUTA: Next: Next
Picture1.Refresh
Z = SNDPLAYSOUND(CurDir$ + "¥start.wav", 1)
REPLAY.Enabled = 0
USU = 0: MV = 0
MousePointer = 1

End Sub


Private Sub NEWG_Click()

Call NEWGAME

End Sub

Private Sub NEWGAME()

Dim F As Integer, G As Integer, T As Integer, P As Integer
Dim X1 As Integer, Y1 As Integer

Randomize
MousePointer = 11: Timer1.Interval = 0
'石の配置
Do
F = 0
For M = 0 To 11: For N = 0 To 11: A(M, N) = 20: C(M, N) = 20: Next: Next
For M = 1 To 10: For N = 1 To 10: C(M, N) = 0: Next: Next
C(5, 5) = 1: C(6, 5) = 1
For T = 2 To KSU
    G = 0
    K = T Mod 9: If K = 0 Then K = 9
    Do
        M = Int(Rnd * 8) + 2: N = Int(Rnd * 8) + 2
        If KSU < 12 Then M = Int(Rnd * 6) + 3: N = Int(Rnd * 6) + 3
        G = G + 1: If G > 100 Then F = 1: Exit Do
        Do
            X1 = Int(Rnd * 2): Y1 = Int(Rnd * 2)
        Loop While Abs(X1 + Y1) <> 1
        If C(M, N) = 0 And C(M + X1, N + Y1) = 0 Then
            If C(M-1, N) > 0 Or C(M+1, N) > 0 Or C(M, N-1) > 0 Or C(M, N+1) > 0 Then
                C(M, N) = K: C(M + X1, N + Y1) = K: Exit Do
            End If
        End If
        DoEvents
    Loop
    If F = 1 Then Exit For
    GoSub SFL
Next
Loop While F = 1

For T = 1 To 30
    Do
        M = Int(Rnd * 10) + 1: N = Int(Rnd * 10) + 1
    Loop While C(M, N) = 0
    GoSub SFL
Next
Call MAKE
TMR = 0: Label1.Caption = "TIME 0": Timer1.Interval = 1000
Exit Sub

'石を移動
SFL:
For P = 1 To 20
    Do
        X1 = 1 - Int(Rnd * 3): Y1 = 1 - Int(Rnd * 3)
    Loop While Abs(X1 + Y1) <> 1
    If C(M - X1, N - Y1) > 0 And C(M - X1, N - Y1) < 10 And C(M, N) <> C(M - X1, N - Y1) And
C(M + X1, N + Y1) = 0 Then
```


★新幹線車内でOF書いたりするのは自分ぐらい？（字がいつにもまして汚いのは，本当に新幹線の中で書いているからです。あとペンがちがう)。(東京都練馬区・ルストフレム14歳)……**【編集長**：この原稿は何だ！**影**：電車の中で入力したもんですから。**編集長**：？？**】**

```
        K = C(M, N): M2 = M: N2 = N
        Do
            M = M + X1: N = N + Y1
        Loop While C(M, N) = 0
        If C(M, N) <> K Then C(M2, N2) = 0: C(M - X1, N - Y1) = K: Exit For
    End If
Next
Return

End Sub
Private Sub Picture1_MouseDown(Button As Integer,Shift As Integer,X As Single,Y As Single)

'一手もどす
If Button = 2 And USU > 0 Then
    Z = SNDPLAYSOUND(CurDir$ + "¥warp.wav", 1)
    K = Val(Left$(UND$(USU), 2))
    M = Val(Mid$(UND$(USU), 3, 2)): N = Val(Mid$(UND$(USU), 5, 2))
    Call PUTA
    M = Val(Mid$(UND$(USU), 7, 2)): N = Val(Mid$(UND$(USU), 9, 2))
    If M > 0 Then Call PUTA
    K = 0
    M = Val(Mid$(UND$(USU), 11, 2)): N = Val(Right$(UND$(USU), 2))
    If M > 0 Then Call PUTA
    Picture1.Refresh
    USU = USU - 1
    Exit Sub
End If

'石の指定
M2 = X ¥ 32 + 1: N2 = Y ¥ 32 + 1
K2 = A(M2, N2)
If K2 = 0 Then Exit Sub
MousePointer = 5
MV = 1

End Sub

Private Sub PUTA()

Select Case K
    '石
    Case 0 To 9
        Z = BitBlt(Picture1.hDC, 32*M-32, 32*N-32, 32, 32, CHARA.hDC, 32*K, 0, &HCC0020)
        A(M, N) = K
    '壊れる
    Case 10
        Z = BitBlt(Picture1.hDC, 32*M-32, 32*N-32, 32, 32, CHARA.hDC, 352, 0, &H8800C6)
        Z = BitBlt(Picture1.hDC, 32*M-32, 32*N-32, 32, 32, CHARA.hDC, 320, 0, &H660046)
        A(M, N) = 0
End Select

End Sub

Private Sub Picture1_MouseMove(Button As Integer,Shift As Integer,X As Single,Y As Single)

Dim F As Integer
Dim K1 As Integer, M1 As Integer, N1 As Integer
Dim X1 As Integer, Y1 As Integer
Dim WTM As Long

If MV = 0 Then Exit Sub

M1 = X ¥ 32 + 1: N1 = Y ¥ 32 + 1
If M1 = M2 And N1 = N2 Then Exit Sub
MV = 0
'石を動かす
X1 = Sgn(M1 - M2): Y1 = Sgn(N1 - N2)
If X1 <> 0 And Y1 <> 0 Then Y1 = 0
MousePointer = 1
If K2 = A(M2 + X1, N2 + Y1) Then
    Z = SNDPLAYSOUND(CurDir$ + "¥don.wav", 1)
    K = 10: M = M2: N = N2: Call PUTA
    M = M + X1: N = N + Y1: Call PUTA
    Picture1.Refresh
    WTM = timeGetTime: Do: Loop While timeGetTime - WTM < 100
    K = 0: M = M2: N = N2: Call PUTA
    M = M2 + X1: N = N2 + Y1: Call PUTA
    USU = USU + 1
    UND$(USU) = Right$(CStr(100 + K2), 2) + Right$(CStr(100 + M2), 2) + Right$(CStr(100 + N2),
2) + Right$(CStr(100 + M), 2) + Right$(CStr(100 + N), 2) + "0000"
ElseIf A(M2 + X1, N2 + Y1) <> 0 Then
    Z = SNDPLAYSOUND(CurDir$ + "¥bu.wav", 1)
Else
    Z = SNDPLAYSOUND(CurDir$ + "¥shu.wav", 1)
    M = M2: N = N2
    Do
        K = 0: Call PUTA
        M = M + X1: N = N + Y1
        K = K2: Call PUTA
        Picture1.Refresh
        WTM = timeGetTime: Do: Loop While timeGetTime - WTM < 10
    Loop While A(M + X1, N + Y1) = 0
    If A(M + X1, N + Y1) > 10 Then
```

★「闘技場」と「萌えプロ」，載ると，より恥ずかしいのはどっちのほうでしょうか。（千葉県柏市・**鈴木有人28歳**）……【**つぐ美**：電車に一緒に乗って恥ずかしい人は影さん。**影**：なに，それ？**つぐ美**：覚えてないの？帰りの電車の中。酔っ払いと意気投合してさぁ…。】

```
        K = 0: Call PUTA: Picture1.Refresh
        Z = SNDPLAYSOUND(CurDir$ + "¥biri.wav", 1)
        MsgBox "外に飛び出した!!", 0 + 48, "ミス"
        Call MAKE: Exit Sub
    End If
    Z = SNDPLAYSOUND(CurDir$ + "¥don.wav", 1)
    If K2 = A(M + X1, N + Y1) Then
        M1 = M: N1 = N: K = 10: Call PUTA
        M = M + X1: N = N + Y1: Call PUTA
        Picture1.Refresh
        WTM = timeGetTime: Do: Loop While timeGetTime - WTM < 100
        M = M1: N = N1: K = 0: Call PUTA
        M = M + X1: N = N + Y1: Call PUTA
        USU = USU + 1
        UND$(USU) = Right$(CStr(100 + K2), 2) + Right$(CStr(100 + M2), 2) + Right$(CStr(100 +
N2), 2) + Right$(CStr(100 + (M)), 2) + Right$(CStr(100 + (N)), 2) + "0000"
    Else
        USU = USU + 1
        UND$(USU) = Right$(CStr(100 + K2), 2) + Right$(CStr(100 + M2), 2) + Right$(CStr(100 +
N2), 2) + "0000" + Right$(CStr(100 + M), 2) + Right$(CStr(100 + N), 2)
    End If
End If
Picture1.Refresh
'クリア
F = 0
For M = 1 To 10
    For N = 1 To 10
        If A(M, N) > 0 Then F = 1: Exit For
    Next
    If F = 1 Then Exit For
Next
If F = 0 Then
    Timer1.Interval = 0
    Z = SNDPLAYSOUND(CurDir$ + "¥tada.wav", 1)
    MsgBox "みごとクリアじゃ!!", 0 + 48, "ファラオの言葉"
    If TMR <= 2 * KSU - 3 Then
        Picture1.Cls
        Picture1.Line (0, 250)-(320, 250), &HFFFFFF
        For N = 1 To 130 Step 2
            Picture1.Line (140 - N, 120 + N)-(140 + N, 120 + N), &HC0C0C0
        Next
        For N = 1 To 80 Step 2
            Picture1.Line (230 - N, 170 + N)-(230 + N, 170 + N), &HFFFFFF
        Next
        KSU = KSU + 1
        MsgBox "次は" + Str$(2 * KSU) + "個に挑戦じゃ!!" + Chr$(13) + Str$(2 * KSU - 3) + "秒以
内にクリアするのじゃ!!", 0 + 48, "ファラオの言葉"
    Else
        MsgBox Str$(2 * KSU - 3) + "秒以内にクリアするのじゃ。", 0 + 48, "ファラオの言葉"
    End If
    Call NEWGAME
    Exit Sub
End If
REPLAY.Enabled = -1

End Sub

Private Sub Picture1_MouseUp(Button As Integer, Shift As Integer, X As Single, Y As Single)

MousePointer = 1
MV = 0

End Sub

Private Sub REPLAY_Click()

Call MAKE

End Sub


Private Sub Timer1_Timer()

TMR = TMR + 1
Label1.Caption = "TIME" + Str$(TMR)

End Sub
```

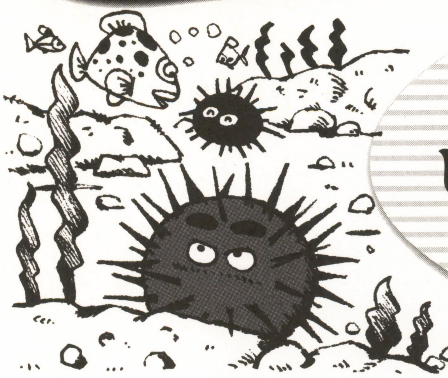
《リスト２》モジュールのプログラム・リスト

```
Declare Function SNDPLAYSOUND% Lib "winmm" Alias "sndPlaySoundA" (ByVal Fn$, ByVal Flag%)
Declare Function BitBlt% Lib "gdi32" (ByVal destDC%, ByVal X%, ByVal Y%, ByVal w%, ByVal H%,
ByVal srcDC%, ByVal xSrc%, ByVal ySrc%, ByVal Rop&)
Declare Function timeGetTime Lib "winmm.dll" () As Long

Public A(11, 11) As Integer, C(11, 11) As Integer
Public K As Integer, M As Integer, N As Integer, Z As Integer
Public K2 As Integer, M2 As Integer, N2 As Integer
Public MV As Integer, KSU As Integer, USU As Integer
Public TMR As Integer
Public UND(200) As String
```


★影さんが，最近パソコンを破壊してくれないので寂しいです。ぜひ，残暑の哀愁を込めた一発をお願いします。先生！（東京都中央区・まぢけん20歳）……【**影**：先生だなんて，うれしい響きだなぁ。もう1回言って！**つぐ美**：先生，お仕事がたくさん残ってますよ。】

Windows(Visual Basic V6.0)用



敵を同士討ちさせるのだ

ウニのさわ君(くん)2001

宮西　和機（15歳）

内容

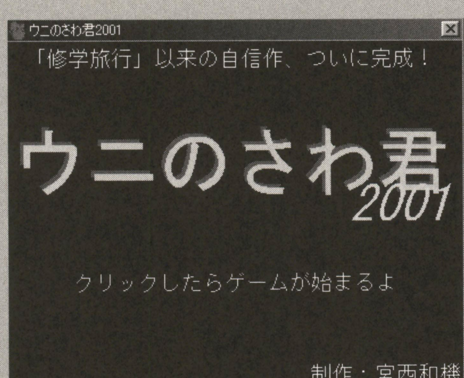
この『ウニのさわ君2001』は，2001年2月号のベーマガに掲載された我が作品である『修学旅行〜三日目の夜〜』を大幅に改良したものです。ゲームシステムなどは前回の作品にとても似ていますが，アルゴリズムなどは凝ったつもりなので，そこを見ていただきたいです。

ストーリー

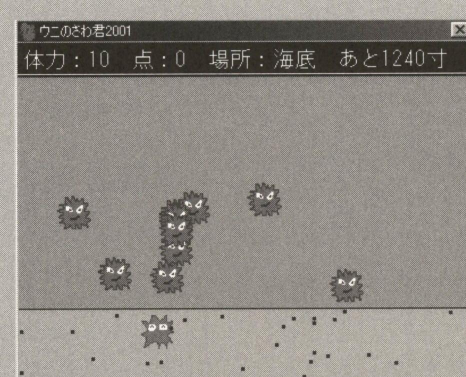
広大な海に1匹のウニがいた。そのウニの名前は「さわ」。さわ君はある日友達から一つの話を聞いた。

その話とは，「この海から陸地に上がればドライブインがある。そのドライブインに行けば，なんとウニ丼にしてくれるのだ。我々ウニにとっては素晴らしいことだ。しかし，そのドライブインに行く経路には，爆発性のある爆弾ウニがたくさん漂っていて，そのウニを避けながら進んでいかないと接触していっしょに爆死してしまう可能性がある。だが，その爆弾ウニ同士を接触させれば，その爆弾ウニを倒すことができる。爆弾ウニを倒しながらドライブインを目指すのだ！」という話だ。

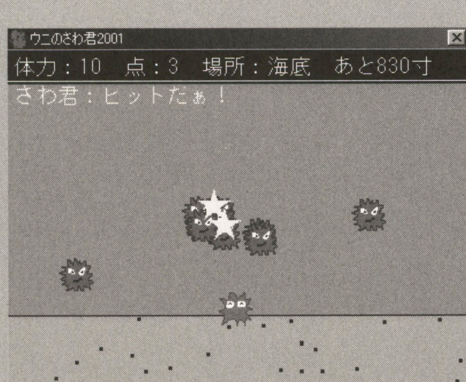
さわ君は期待と不安を入り混じらせながらドライブインへ行くことにした。



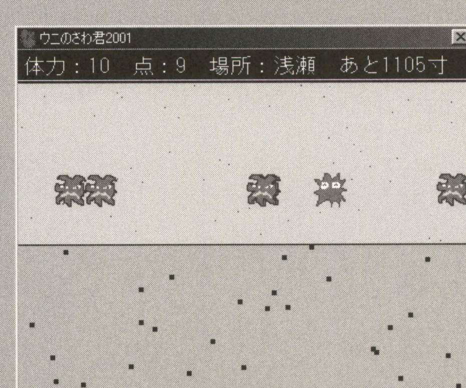
《写真1》ウニが主役のゲーム



《写真2》近づいてくる敵をかわしながら進む



《写真3》敵は同士討ちさせることで倒していく

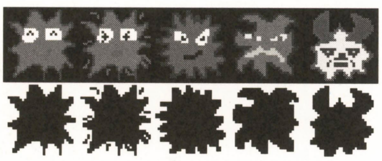


《写真4》先へ進むごとに，場所や敵キャラが変化する

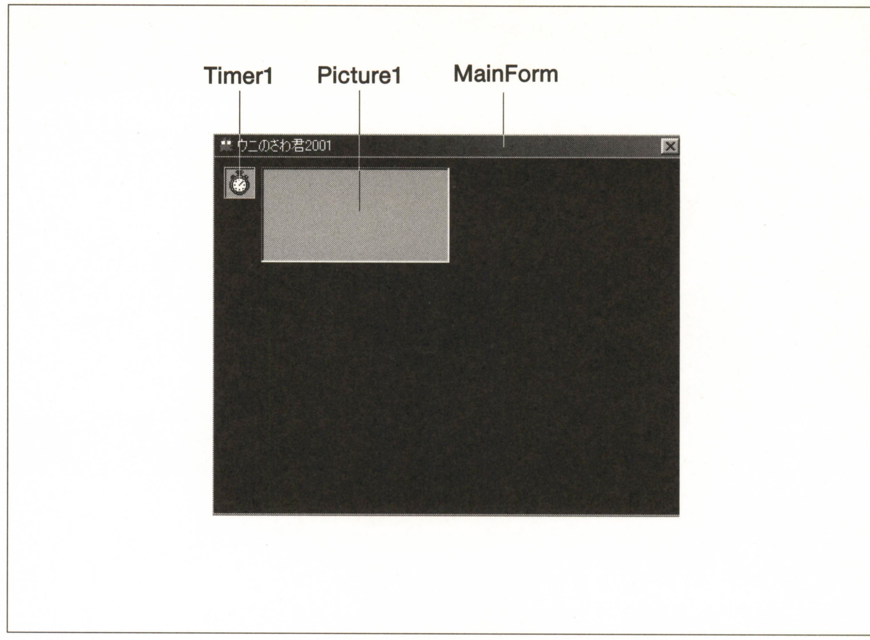
★弟のパソコン部の仲間に「BASIC STUDIO」を買った奴がおる！と聞き，8月号までを持って行かせた。9月号は自分で買ってくれただろうか。…3年生ってトコがちょっと気になるが，いい趣味じゃないか。（愛知県名古屋市・わいち301　27歳）……【Dr.D：うむ。】

《第１表》変数表

i,j	forステートメント用の変数
Game	ゲームモード
wa	ウェイト，ゲーム進行用
MyHP	さわ君の体力
MyX,MyY	さわ君の座標
Sco	得点
stage	場所
HEne	爆弾ウニの出現数
HBom	爆発表現の数
StaNa(0-2)	場所の名前
dma	ダメージを受けたときのウェイト
Time	目的地までの距離
spe	爆弾ウニの移動速度
mc	さわ君の喋った言葉の色
Ene(0-15)	爆弾ウニのデータ
Ishi(0-29)	背景描画用のデータ
p(0-49)	背景描画用のデータ
Bom(0-2)	爆発表現のデータ
MyHPS	さわ君の体力の文字列
ScoS	得点の文字列
TimeS	目的地までの距離の文字列



unisawa.bmpの画像



フォームの配置図

《第２表》プロパティ表

フォームの設定	
Name	MainForm
AutoRedraw	-1 'True
BackColor	&H00000000&
BorderStyle	1 '固定(実線)
Caption	ウニのさわ君2001
Font	ＭＳ ゴシック
Height	4875
Icon	(アイコン)
Left	0
LinkTopic	Form1
MaxButton	0 'False
MinButton	0 'False
ScaleHeight	300
ScaleMode	3 'ピクセル
ScaleWidth	400
StartUpPosition	2 '画面の中央
Top	0
Width	6090

ピクチャボックスの設定	
Name	Picture1
AutoRedraw	-1 'True
Font	ＭＳ 明朝
Height	81
Left	40
ScaleHeight	1155
ScaleWidth	2355
TabIndex	0
Top	120
Visible	0 'False
Width	161

タイマーの設定	
Name	Timer1
Left	8
Top	8

遊びかた

基本的に操作はマウス操作のみです。

タイトル画面のときに画面のどこかをクリックすればゲームが始まります。

主人公であるさわ君はマウスカーソル位置と同じになります。ゲームが始まってしばらくすると画面のはじから敵が接近してきます。敵はさわ君のほうへひたすら接近してくるのでそれを利用し，敵を誘導してください。敵同士が完全に重なると爆発します。それにより得点を上げていきます。

しかし，早く得点をかせがないと，つぎの目的地にすぐ着いてしまうので要注意です。

ステージは全部で三つあり，体力が0になると敗北。完全クリアそして高得点を狙ってドライブインをめざそう。

操作方法

主にマウスを使用。クリックはゲームを進行させるときのみ使用し，ゲーム中は使用しません。

ゲーム中にマウスを移動させるとマウスカーソルと同じ位置にさわ君がついてきます。

改造のアドバイス

コンピュータの速度に合わせてタイマーコントロールのIntervalプロパティを設定してみてください。

それと，BGMや効果音をつけるといいかも。音の鳴らしかたは色々な方法があるから，ゲームの動作環境やコーディングの面で色々と試してみてください。



by 都田 祥

評…そうか，まだクリアしていないのね（編）

CHECKER FLAG

影：説明文どおり，画面表示が大幅に変わってはいるが，基本システムは『修学旅行～三日目の夜～』と同じ。もう少ししっかりした改良がほしかったかな。

Dr.D：全体的な流れはうまくまとまっている感じじゃが，このゲームのポイントとなる敵キャラの動きが，if文の組み合わせという基本的な形式なので，凝っているとは言い難い。こだわった形にするなら，ゲームのポイントとなる動きの処理に凝ってほしい。この敵キャラの動きでいえば，ベクトル計算などを取り入れたホーミング処理などを実現してほしいものじゃ。

編：こういった続編の投稿は，前回と明らかに違う点を明確に打ちだしてほしいですね。リスト面の改良だけでは，普通に遊んでいる側から見ると何も変化していないと感じますから。

★2時間かかって組んだプログラムをわずか1秒で消し去る「強制終了」…ああ恐ろしや。（青森県南津軽郡・ていこく22歳）……【つぐ美：たしかにそうよね。プログラムもワープロもなんでもコマ目に保存しておきましょうね。あっ，影さんだ。原稿の保存，保存と。】

苦労したところ

敵の行動プログラムです。速度を速くすると，さわ君に近づいたときに変な行動をおこすエラーを直すのに一苦労しました。

あとがき

この投稿コーナーに載っているプログラムは，今これを読んでいるあなたの開発環境に打ち込んでやりましょう。今はインターネットでダウンロードできちゃいますけど，僕はいっぱい打ち込んで学習してきました。打ち込むと，タイピングがうまくなる。開発環境に慣れてくる。そして，打ち込むことによってそのプログラムの内容が把握しやすくなり改造もしやすくなって，そのプログラムの中で使っている技法なんかも理解できちゃうという素晴らしい利点があります。さぁ，ベーマガに定規を置き，今どの行をを打っているのか確認しながら楽しいプログラミングライフを過ごしましょう。

《リスト１》『ウニのさわ君2001』のプログラム・リスト

```
Option Explicit
'BitbltAPIの宣言（画像ブロック転送用ＡＰＩ）
Private Declare Function BitBlt Lib "gdi32" (ByVal hDestDC As Long, ByVal x As Long, ByVal y As Long, ByVal nWidth As Long, ByVal nHeight As Long, ByVal hSrcDC As Long, ByVal xSrc As Long, ByVal ySrc As Long, ByVal dwRop As Long) As Long

'変数の宣言
Dim i%, j%
Dim Game%, wa%, MyHP%, MyX%, MyY%, Sco%, stage%
Dim HEne%, HBom%, StaNa$(2), dma%, Time%, spe%, mc As Long
Dim Ene(15) As Mono, Ishi(29) As Mono, p(49) As Mono, Bom(2) As Bomb
Dim MyHPS$, ScoS$, TimeS$

'物体の座標を表す構造体の定義
Private Type Mono
    x As Integer
    y As Integer
End Type

'爆発表現を表す構造体の定義
Private Type Bomb
    x As Integer
    y As Integer
    wa As Integer
End Type

Private Sub Form_Load()
    Randomize Timer
    Picture1.Picture = LoadPicture(App.Path & "¥unisawa.bmp")
    Call Reset
    Timer1.Interval = 1
End Sub

Private Sub DT(x As Integer, y As Integer, str As String, sz As Integer, c As Long)

    'DrawText  フォームに文字を描きます
    With MainForm
        .FontSize = sz
        .ForeColor = c
        .CurrentX = x
        .CurrentY = y
    End With
    MainForm.Print str
End Sub

Private Sub Form_MouseDown(Button As Integer, Shift As Integer, x As Single, y As Single)
    If wa = 0 Then
        wa = 1 'ゲームモードをプレイモードにする
    ElseIf wa = 37 Then
        wa = 1
        stage = stage + 1
        HEne = 0
        dma = 0
        For i = 0 To 2
            Bom(i).wa = 0
        Next i
        Select Case stage
            Case 1
                Time = 1500
                spe = 2
                mc = vbBlack
            Case 2
                Time = 1000
                spe = 3
                mc = vbYellow
        End Select
    ElseIf wa = 38 Or wa = 39 Then
        wa = 0
        Call Reset
    End If
End Sub

Private Sub Reset()

    '変数の初期化
    Game = 0    'ゲームモードをタイトルモードにする
    wa = 0
    MyHP = 10   '自キャラのＨＰを最大にする
    Sco = 0     'スコアを初期化にする
    stage = 0   '場所を海底（最初）にする
    HEne = 0    '敵の数を初期化する
    HBom = 0    '爆発表現数を初期化する
    StaNa(0) = "海底"
    StaNa(1) = "浅瀬"
    StaNa(2) = "陸地"
    dma = 0
    Time = 2000 'ゴールまでの距離を設定
    spe = 1     '敵のスピードを設定
    mc = vbYellow 'さわ君のセリフの色
    For i = 0 To 2
        Bom(i).wa = 0
    Next i
    '文字列の長さを設定
    MyHPS = "01"
    ScoS = "012"
    TimeS = "0123"
End Sub

Private Sub Form_MouseMove(Button As Integer, Shift As Integer, x As Single, y As Single)
    MyX = x: MyY = y
    If MyY < 46 Then MyY = 46
End Sub

Private Sub Timer1_Timer()
    MainForm.Cls
    If wa = 0 Then

        'タイトルの描画
        With MainForm
        DT 15, 5, "「修学旅行」以来の自信作、ついに完成！", 14, vbWhite
        DT 2, 70, "ウニのさわ君", 48, &H8000&
        DT 5, 73, "ウニのさわ君", 48, vbGreen
        .FontItalic = True
        DT 300, 120, "2001", 32, vbCyan
        .FontItalic = False
        DT 57, 200, "クリックしたらゲームが始まるよ", 14, vbYellow
        DT 265, 280, "制作：宮西和機", 14, vbWhite
        End With
```

★9月号のパソコン・レクチャーや巻末にモールス信号の話が…。くそう，私の脳内がノールス信号でなかったなら！（愛知県名古屋市・ズオウ・リアキ29歳）……【影：ツートントントン，ツートン。編：また宇宙人と連絡取り合ってるとか言うんですよ。影：にやり。】

```
ElseIf wa > 0 And wa < 25 Then
    DT 100, 95, "準備", 72, RGB(200, 200, 255)
    wa = wa + 1
ElseIf wa >= 25 And wa < 35 Then
    MainForm.MousePointer = 2
    DT 100, 95, "開始", 72, RGB(255, 200, 255)
    wa = wa + 1
ElseIf wa = 35 Then

    '背景の部品をランダムに配置
    Select Case stage
        Case 0
            For i = 0 To 29
                Ishi(i).x = Int(Rnd * 400): Ishi(i).y = Int(Rnd * 68) + 232
            Next i
            For i = 0 To 49
                p(i).x = Int(Rnd * 400): p(i).y = Int(Rnd * 200) + 30
            Next i
        Case 1
            For i = 0 To 29
                Ishi(i).x = Int(Rnd * 400): Ishi(i).y = Int(Rnd * 128) + 172
            Next i
            For i = 0 To 49
                p(i).x = Int(Rnd * 400): p(i).y = Int(Rnd * 140) + 30
            Next i
        Case 2
            For i = 0 To 29
                Ishi(i).x = Int(Rnd * 400): Ishi(i).y = Int(Rnd * 48) + 252
            Next i
            For i = 0 To 49
                p(i).x = Int(Rnd * 400): p(i).y = Int(Rnd * 220) + 30
            Next i
    End Select
    wa = 36
ElseIf wa = 36 Then

    '背景の部品を左に移動させていく
    For i = 0 To 29
        Ishi(i).x = Ishi(i).x - 3
        If Ishi(i).x < -3 Then
            Ishi(i).x = 403
            Select Case stage
                Case 0
                    Ishi(i).y = Int(Rnd * 68) + 232
                Case 1
                    Ishi(i).y = Int(Rnd * 128) + 172
                Case 2
                    Ishi(i).y = Int(Rnd * 48) + 252
            End Select
        End If
    Next i
    For i = 0 To 49
        p(i).x = p(i).x - 2
        If p(i).x < -1 Then
            p(i).x = 401
            Select Case stage
                Case 0
                    p(i).y = Int(Rnd * 200) + 30
                Case 1
                    p(i).y = Int(Rnd * 140) + 30
                Case 2
                    p(i).y = Int(Rnd * 220) + 30
            End Select
        End If
    Next i

    '敵を時々出現させる
    If Int(Rnd * 50) = 0 And HEne < 16 Then
        HEne = HEne + 1
        EneReset HEne - 1
        End If
```

```
    '敵を自キャラに近づくように移動させる
    If HEne > 0 Then
        For i = 0 To HEne - 1
            If Ene(i).x < MyX And (MyX - Ene(i).x) >= spe Then
                Ene(i).x = Ene(i).x + spe
            ElseIf Ene(i).x < MyX And (MyX - Ene(i).x) <= spe Then
                Ene(i).x = Ene(i).x + 1
            ElseIf Ene(i).x > MyX And (Ene(i).x - MyX) >= spe Then
                Ene(i).x = Ene(i).x - spe
            ElseIf Ene(i).x > MyX And (Ene(i).x - MyX) <= spe Then
                Ene(i).x = Ene(i).x - 1
            End If
            If Ene(i).y < MyY And (MyY - Ene(i).y) >= spe Then
                Ene(i).y = Ene(i).y + spe
            ElseIf Ene(i).y < MyY And (MyY - Ene(i).y) <= spe Then
                Ene(i).y = Ene(i).y + 1
            ElseIf Ene(i).y > MyY And (Ene(i).y - MyY) >= spe Then
                Ene(i).y = Ene(i).y - spe
            ElseIf Ene(i).y > MyY And (Ene(i).y - MyY) <= spe Then
                Ene(i).y = Ene(i).y - 1
            End If
        Next i
    End If

    '敵同士の衝突判定
    For i = 0 To HEne - 1
        For j = 0 To HEne - 1
            If i <> j Then
                If Ene(i).x = Ene(j).x And _
                   Ene(i).y = Ene(j).y Then
                    Sco = Sco + 1
                    Bom(HBom).x = Ene(i).x
                    Bom(HBom).y = Ene(i).y
                    EneReset i: EneReset j
                    Bom(HBom).wa = 25
                    HBom = HBom + 1
                    If HBom = 3 Then HBom = 0
                End If
            End If
        Next j
    Next i

    '自キャラと敵の衝突判定
    For i = 0 To HEne - 1
        If MyX - 8 < Ene(i).x + 8 And _
           MyX + 8 > Ene(i).x - 8 And _
           MyY - 8 < Ene(i).y + 8 And _
           MyY + 8 > Ene(i).y - 8 And _
           dma = 0 Then
            dma = 25
            MyHP = MyHP - 1
            EneReset i
            If MyHP = 0 Then
                wa = 38 '敗北する
            End If
        End If
    Next i

    'ステージ制御
    Time = Time - 1
    If Time = 0 Then
        If stage = 2 Then
            wa = 39
        Else
            wa = 37
        End If
    End If
    Call GameDraw
ElseIf wa = 37 Then
    DT 70, 40, "得点は  " & Sco, 14, vbWhite
    DT 100, 95, "進行", 72, RGB(250, 150, 50)
```

★矢倉は将棋の純文学。RPGはゲームの純文学。（千葉県山武郡・**上野潤之助14歳**）……【**影**：Dr.，将棋で勝負！おっDr.は矢倉で陣立てしてきましたね。では私の新しい作戦を。名づけて“グラグラ”とでも申しましょうか。**Dr.D**：なんでもいいから，早く打てよ。】

```
            DT 57, 200, "クリックしたらゲームがつづくよ", 14, vbYellow
        ElseIf wa = 38 Then
            DT 70, 40, "得点は  " & Sco, 14, vbWhite
            DT 100, 95, "敗北", 72, RGB(250, 150, 50)
            DT 57, 200, "クリックしたらタイトルにいくよ", 14, vbYellow
        ElseIf wa = 39 Then
            DT 70, 40, "得点は  " & Sco, 14, vbWhite
            DT 100, 95, "完了", 72, vbRed
            DT 57, 200, "クリックしたらタイトルにいくよ", 14, vbYellow
        End If
End Sub

Private Sub GameDraw()

    'ゲーム画面の描画
    RSet MyHPS = MyHP
    RSet ScoS = Sco
    RSet TimeS = Time
    DT 5, 5, "体力：" & MyHPS & "  点：" & ScoS & "  場所：" & StaNa(stage) & "  あと" & TimeS
& "寸", 14, vbWhite
    MainForm.Line (0, 27)-(400, 27), vbWhite
    Select Case stage
        Case 0 '海底の描画
            MainForm.Line (0, 30)-(400, 230), &HC0C000, BF
            MainForm.Line (0, 232)-(400, 300), &HFFC0C0, BF
            For i = 0 To 29
                MainForm.Line (Ishi(i).x,Ishi(i).y)-(Ishi(i).x+2,Ishi(i).y+2),vbBlack,BF
            Next i
            For i = 0 To 49
                MainForm.PSet (p(i).x, p(i).y), vbRed
            Next i
        Case 1 '浅瀬の描画
            MainForm.Line (0, 30)-(400, 170), &HFFFF00, BF
            MainForm.Line (0, 172)-(400, 300), &HC0C0FF, BF
            For i = 0 To 29
                MainForm.Line (Ishi(i).x,Ishi(i).y)-(Ishi(i).x+3,Ishi(i).y+3),vbBlack,BF
            Next i
            For i = 0 To 49
                MainForm.PSet (p(i).x, p(i).y), vbBlue
            Next i
        Case 2 '陸地の描画
            MainForm.Line (0, 30)-(400, 250), vbBlack, BF
            MainForm.Line (0, 252)-(400, 300), &H4080&, BF
            For i = 0 To 29
                MainForm.Line (Ishi(i).x,Ishi(i).y)-(Ishi(i).x+5,Ishi(i).y+5),&HE0E0E0,BF
            Next i
            For i = 0 To 49
                MainForm.PSet (p(i).x, p(i).y), vbWhite
            Next i
    End Select
    If dma = 0 Then
        BitBlt MainForm.hDC, MyX - 16, MyY - 16, 32, 32, Picture1.hDC, 0, 32, vbSrcAnd
        BitBlt MainForm.hDC, MyX - 16, MyY - 16, 32, 32, Picture1.hDC, 0, 0, vbSrcPaint
    ElseIf dma Mod 2 = 0 Then
        BitBlt MainForm.hDC, MyX - 16, MyY - 16, 32, 32, Picture1.hDC, 32, 32, vbSrcAnd
        BitBlt MainForm.hDC, MyX - 16, MyY - 16, 32, 32, Picture1.hDC, 32, 0, vbSrcPaint
    End If
    If HEne > 0 Then
        For i = 0 To HEne - 1
            BitBlt MainForm.hDC, Ene(i).x - 16, Ene(i).y - 16, 32, 32, Picture1.hDC, 64 + stage
* 32, 32, vbSrcAnd
            BitBlt MainForm.hDC, Ene(i).x - 16, Ene(i).y - 16, 32, 32, Picture1.hDC, 64 + stage
* 32, 0, vbSrcPaint
        Next i
    End If
    For i = 0 To 2
        If Bom(i).wa > 0 Then
            DT Bom(i).x - 16, Bom(i).y - 16, "★", 24, vbYellow
            DT 5, 30, "さわ君：ヒットだぁ！", 14, mc
            Bom(i).wa = Bom(i).wa - 1
        End If
    Next i
    If dma > 0 Then
        Select Case dma
            Case 25: MainForm.Top = MainForm.Top + 10
            Case 24: MainForm.Left = MainForm.Left - 10
            Case 23: MainForm.Top = MainForm.Top - 20
            Case 22: MainForm.Left = MainForm.Left + 20
            Case 21: MainForm.Top = MainForm.Top + 20
            Case 20: MainForm.Left = MainForm.Left - 10
            Case 19: MainForm.Top = MainForm.Top - 10
        End Select
        DT 5, 55, "さわ君：うわぁ！", 14, mc
        dma = dma - 1
    End If
    If Time < 50 Then
        DT 5, 80, "さわ君：目的地に着いたぞ！", 14, mc
    End If
End Sub

Private Sub EneReset(n As Integer)

    '敵をランダムに配置
    If Int(Rnd * 2) = 0 Then
        Ene(n).x = Int(Rnd * 100) - 130
    Else
        Ene(n).x = Int(Rnd * 100) + 430
    End If
    Ene(n).y = Int(Rnd * 238) + 48
End Sub
```



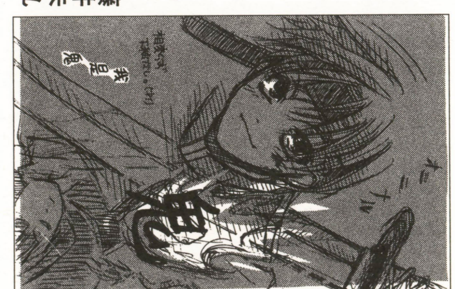
評…ほのぼのして好きですこの絵（つぐ美）

by 和々草



評…モンゴルの大草原を走っているそうです（影）

by 春陽



評…さぁ、仕事仕事（編集長）　鬼！（影）

by 藤井朱丸



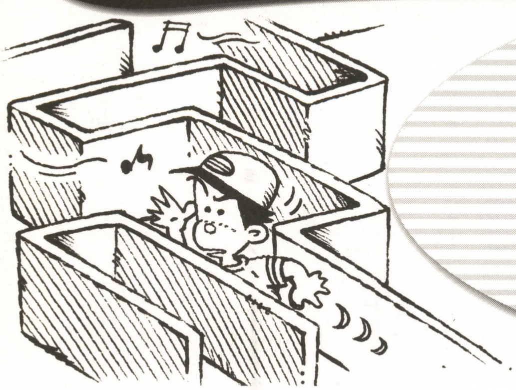
評…GBA万歳のコメントあり。念のため（影）

by ARAI

★「ラ製」が休刊したとは知らなかった。最近のベーマガに電子工作の記事が多いのはそのため？（岩手県和賀郡・砂漠の海賊！パニッシュモンスター29歳）……【編：自分で何かを創り出すクリエイティブ性を養う点は，プログラムも電子工作も同じですしね。】

(超大作プログラム)　リストが誌面に掲載できない作品です。
BMホームページ(http://www.basicmagazine.net)からダウンロードしてお楽しみください。

Windows(Visual Basic V6.0 + DirectX7)用



音をヒントに3D迷路を攻略だ

迷路（めいろ）

橘　一馬（29歳）

あいさつ

3Dの迷路が作りたいと思って，いろいろと他のプログラムを見たのですが，よくわからなくて「3DなんだからDirect3Dをつかってみよう！」と思っちゃって作っちゃいました。

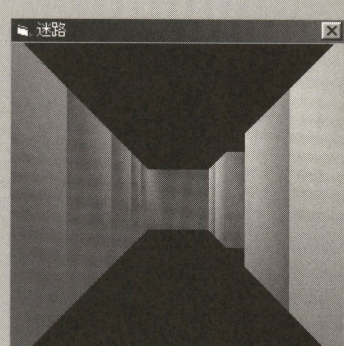
はじめてDirect3Dをつかってプログラミングしたゲームです。だから，よくわかっていないでプログラミングしてしまいました。いろいろとご指導お願いします。これからも3Dゲームを作るぞ！！

遊びかた

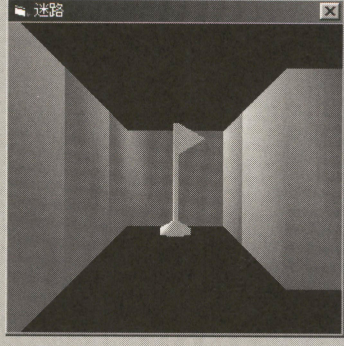
タイトルでリターン・キーを押すと始まります。カーソルで移動して旗を見つけてください。スペース・キーを押すとジャンプします。音がヒントになってます。制限時間は1分です。僕は5面ぐらいしか行きません。



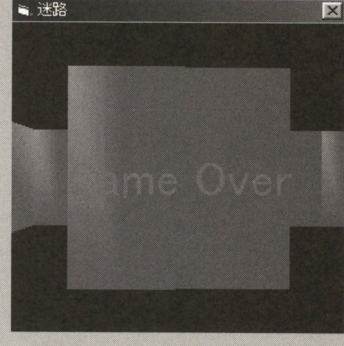
《写真1》3Dと音の表現がウリの迷路



《写真2》3D表現の迷路は独特の雰囲気がある



《写真3》この旗のところがゴールになる



《写真4》1分以内にゴールできないとゲーム・オーバー

CHECKER FLAG

編：タイトルどおりのゲーム内容ですが，3D形式なので見栄えのするゲームに仕上がっています。

つぐ美：また，本誌の写真からは当然わからないことですが，じつはこのゲーム，音をヒントにゴールの旗を見つけていきます。旗の位置によって音の発生位置や音量などが変化するのです。

Dr.D：このゲームのウリは，3D形式ということよりも，つぐ美クンが述べている音の部分じゃろう。DirectXというと，DirectDrawやDirect3Dに注目しがちじゃが，Sound面の機能もアイデア次第で，ゲームを構成する重要な要素になりうるのじゃ。

★参考にしたもの

- 「Visual BasicによるDirectX7　活用術」　川田徹 著
- 「DirectX7 For Visual Basic　はじめるゲームプログラミング」　藤田伸二 著



by 鈴木有人

評…わーい、カキ氷！小倉がおすすめ（つぐ美）

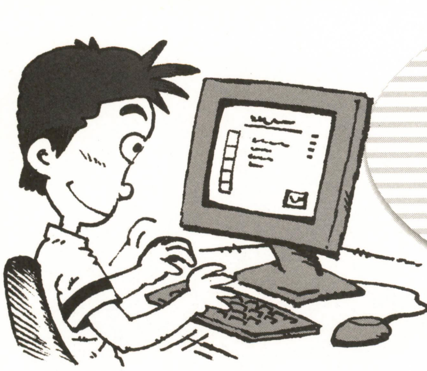
★1日1回体重計。（茨城県古河市・年齢不相応41歳）……【つぐ美：私に何か言いたいようですね。影：このハガキを採用したぐらいで，そこまで考えるのもおかしいよ。つぐ美：そ，そうよね。影：そうそう体重の2kgや10kg気にしないこと。つぐ美：ボカーン。】

Windows(Visual C++ V6.0)用

Visual CでBASIC形式プログラムを

実用

BCBC（ビーシービーシー）クラス&（アンド）サンプル



BlueCube（19歳）

内容

BCBCとは，BlueCube Basic charactical classの略称です。なんとなく思いつきでBasicっぽい画面を作ってみたくなったので作ってみました。

クラスなので，いろんなプログラムにも簡単に組み込めます。っていうか，クラスがメインなので，サンプルはわかりやすいように簡単なものにしたので，おもしろくないです。

まぁ，ベーマガを読んでる人ならこれを最大限活用したおもしろいものを作ってくれると信じてます。

使いかた

まずはプログラムのフォルダに「bcbcc.h」を入れます。ソース内ではこれをインクルード（#include"bcbcc.h"）してください。

つぎにグローバル変数として，BCBC *cs;を宣言します。

そしてWM_CREATEで，cs=new BCBC(mdc,640,480);と入れてください（mdcのところはメモリDC，640，480はその幅と高さ）。

あと，WM_DESTROYで必ず開放（delete cs;）してやってください。

あとの使いかたは，bcbcc.hファイル内に記載しておきます。

サンプルプログラムの操作方法

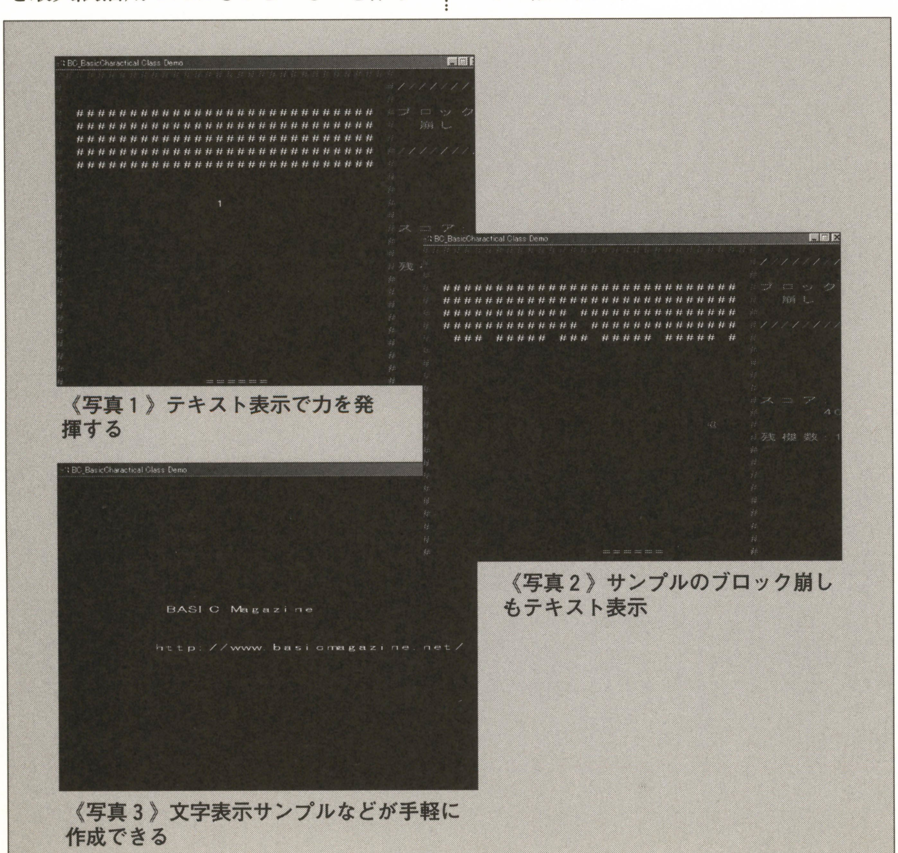
タイトルのとき，スペース・キーでスタートします。カーソル・キーの左右でバーを移動できます。スペース・キーを押すとバーが上下します。

まぁ，普通のブロック崩しなのでボールを落とさないようにしてください。

今回はサンプルのため，ボールの動きが適当なので，難しいかもしれません。

プログラムの仕組み

bcwin.hには，あると結構便利な関数



《写真1》テキスト表示で力を発揮する

《写真2》サンプルのブロック崩しもテキスト表示

《写真3》文字表示サンプルなどが手軽に作成できる

CHECKER FLAG

つぐ美： サンプルゲームの画面のような，テキスト表示をVisual C++で手軽に扱えるようになるプログラムです。ゲーム自体は単純なもので，面白味には欠けるけど，このプログラムの使いかたを知るという面で，このようなサンプルは重要ですよね。

Dr.D： こういったクラスという形での実用プログラムというのも，新しい流れの投稿じゃな。内容としては，そんな難しいことを定義してはおらんが，BASICライクで手軽にプログラムできるので，チョットした表示をしたいときなどに使えるクラスと言えるじゃろう。

影： 過去のテキスト表示の投稿プログラムの移植なんかには，重宝しそうだな。

★おひさ～。ところで，8月号73ページのOFのことで，あれは自称であって，本名は「うさだヒカル」です。（愛媛県今治市・カラプス16歳）……【影：うわーっ，情報が入り乱れて，何がなんだかー。編：影さんの髪の毛みたいですね。影：どういうこっちゃ。】

が入ってます。bcbcc.hはBCBCクラスが入ってます。main.cppはメインのプログラムです。

Basicから C に移ろうとしている人には，少し難しいかもしれませんが，ある程度やっている人には結構わかりやすいはずです。

クラスは使い始めると，とても便利なものなので，ぜひ使いましょう。

改造のアドバイスと苦労した点

昔懐かしい画面が簡単に作れると思いますので，これを使って何か作ってベーマガに投稿してくれたらうれしいです。

ちなみに，ゲームとか作ってる人は変数のモニタとかに使うのに便利です。

苦労したところは，locateの位置あわせですね。最初はprintのとき，文字列を一気にやってたんだけど，それでは位置がずれることがわかって１文字ずつにしました。けど，それだと全角に対応できないから，そこをできるようにするところが大変でした。

《リスト１》bcbcc.hのプログラム・リスト

```
/*
	BCBCクラスの使い方

	コンストラクタ
	BCBC(HDC hdc,int w,int h)

		HDC dc	:	表示用デバイス コンテキストのハンドル
		int w	:	dcの幅
		int h	:	dcの高さ

		まず、これを呼び出します。

	locate(int x,int y)
		int x	:	カーソルのｘ座標
		int y	:	カーソルのｙ座標

		BasicでいうLocateと同じです。
		widthで指定した値まで有効です。

	color(BYTE r,BYTE g,BYTE b);
		BYTE r	:指定色の赤成分
		BYTE g	:指定色の緑成分
		BYTE b	:指定色の青成分

		basicのcolorにあたる関数です。
		文字の色を設定します。

	color(COLORREF col);
		COLORREF col	:	指定色

		basicのcolorにあたる関数です。
		文字の色を設定します。

	print(char c[]);
		char c[]	:	出力文字列

		Basicのprintにあたる関数です。
		locateで指定された位置に文字を出力します。

	print(int x,int y,char c[]);
		int x	:	カーソルのｘ座標
		int y	:	カーソルのｙ座標
		char c[]:	出力文字列

		これは、locateと上のprintをあわせた関数です。

	font(char c[]);
		char c[]	:フォント名

		文字のフォントを指定します。

	width(int w,int h);
		int w	:ｘ方向の文字数
		int h	:ｙ方向の文字数

	Basicのwidthと同じようなものです。
	しかし、basicとは違い好きな値を入れられます。
	一画面分の縦横の文字数を指定できます。

*/
#include <mbstring.h>

class BCBC
	{
	private:
		HDC dc;
		int sw,sh;
		int mw,mh;
		int cx,cy;
		LOGFONT lf;
		HFONT mfont;
		void SetFont(void);
	public:
		BCBC(HDC hdc,int w,int h);
		~BCBC();
		void locate(int x,int y);
		void color(BYTE r,BYTE g,BYTE b);
		void color(COLORREF col);
		void print(char c[]);
		void print(int x,int y,char c[]);
		void font(char c[]);
		void width(int w,int h);
	};

void BCBC::SetFont(void)
	{
	mfont=CreateFontIndirect(&lf);
	SelectObject(dc,mfont);
	return;
	}

BCBC::BCBC(HDC hdc,int w,int h)
	{
	dc=hdc;
	sw=w;
	sh=h;
	cx=0;
	cy=0;
	mw=8;
	mh=16;
	SetBkMode(dc,TRANSPARENT);
	lf.lfHeight			=mh;//(0でデフォルト)
	lf.lfWidth			=mw;//(0でifHeightに合わせる)
	lf.lfEscapement		=0;//角度
	lf.lfOrientation	=0;//角度
	lf.lfWeight			=0;//太さ(0でデフォルト)
	lf.lfItalic			=false;
	lf.lfUnderline		=false;
	lf.lfStrikeOut		=false;
	lf.lfCharSet		=SHIFTJIS_CHARSET;
	lf.lfOutPrecision	=OUT_DEFAULT_PRECIS;
	lf.lfClipPrecision	=CLIP_DEFAULT_PRECIS;
	lf.lfQuality		=DEFAULT_QUALITY;
	lf.lfPitchAndFamily	=DEFAULT_PITCH;
```

★Linuxサイコー！入れたその日からプログラミングできるし，しかもタダ。ちなみに僕はTurboLinux6.0を使っています。（秋田県横手市・**ハングンマル20歳**）……**【影：Linuxは，フィンランドの学生が個人で作ったもので，インターネットで世界に広まったんだ。】**

```
		lstrcpy(lf.lfFaceName,"¥0");
		SetTextColor(dc,RGB(255,255,255));
		}

BCBC::~BCBC()
		{
		DeleteObject(mfont);
		}

void BCBC::locate(int x,int y)
		{
		cx=x;
		cy=y;
		}

void BCBC::color(BYTE r,BYTE g,BYTE b)
		{
		SetTextColor(dc,RGB(r,g,b));
		return;
		}

void BCBC::color(COLORREF col)
		{
		SetTextColor(dc,col);
		return;
		}

void BCBC::print(char c[])
		{
		unsigned char tt[3];
		char t[3];
		for(int i=0;i<(int)strlen(c);i++)
			{
			t[0]=tt[0]=c[i];
			t[1]=tt[1]=c[i+1];
			t[2]=tt[2]='¥0';
			if(_mbsnccnt( tt, 2 )==1 && i!=(int)strlen(c)-1 && t[0]!=' ')
				{
				TextOut(dc,(cx+i)*mw,cy*mh,t,2);
				i++;
				}
			else
				TextOut(dc,(cx+i)*mw,cy*mh,t,1);
			}
		return;
		}

void BCBC::print(int x,int y,char c[])
		{
		locate(x,y);
		print(c);
		return;
		}

void BCBC::font(char c[])
		{
		lstrcpy(lf.lfFaceName,c);
		SetFont();
		return;
		}

void BCBC::width(int w,int h)
		{
		mw=(int)sw/w;
		mh=(int)sh/h;
		lf.lfWidth=mw-5;
		lf.lfHeight=mh-5;
		SetFont();
		return;
		}
```

《リスト2》bcwin.hのプログラム・リスト

```
void SetClientSize(HWND hWnd,int x,int y);
void SetCenter(HWND hWnd);
void CreateCDC(HWND hWnd,HDC &hMemDC,int w,int h);

void SetClientSize(HWND hWnd,int x,int y)
		{
		RECT crc,wrc;
		int ww,wh,cw,ch,w,h;
		GetWindowRect(hWnd,&wrc);
		GetClientRect(hWnd,&crc);
		ww=wrc.right-wrc.left;
		wh=wrc.bottom-wrc.top;
		cw=crc.right-crc.left;
		ch=crc.bottom-crc.top;
		w=ww+x-cw;
		h=wh+y-ch;
		MoveWindow(hWnd,wrc.left,wrc.top,w,h,true);

		/*GetClientRect(hWnd,&crc);
		cw=crc.right-crc.left;
		ch=crc.bottom-crc.top;
		char s[20];
		wsprintf(s,"%d x %d",cw,ch);
		MessageBox(hWnd,s,"",MB_OK);*/
		return;
		}

void SetCenter(HWND hWnd)
		{
		HWND hDWnd;
		RECT drc,rc;
		int x,y,w,h;
	hDWnd = GetDesktopWindow();
	GetWindowRect(hDWnd,&drc);
	GetWindowRect(hWnd,&rc);
	x=(drc.right-(rc.right-rc.left))/2;
	y=(drc.bottom-(rc.bottom-rc.top))/2;
		w=rc.right-rc.left;
		h=rc.bottom-rc.top;
	SetWindowPos(hWnd,HWND_TOP,x,y,w,h,SWP_SHOWWINDOW);
		return;
	}

void CreateCDC(HWND hWnd,HDC &hMemDC,int w,int h)
		{
		void *t;
		BITMAPINFO *bmi;
		char *pBit;
		HBITMAP hbmp;
		bmi=(BITMAPINFO*)malloc(sizeof(BITMAPINFO));
		ZeroMemory(bmi,sizeof(BITMAPINFOHEADER));
		bmi->bmiHeader.biSize=sizeof(BITMAPINFOHEADER);
		bmi->bmiHeader.biWidth=w;
		bmi->bmiHeader.biHeight=h;
		bmi->bmiHeader.biPlanes=1;
		bmi->bmiHeader.biBitCount=24;
		bmi->bmiHeader.biCompression=BI_RGB;
		t=(void*)pBit;
		hbmp=CreateDIBSection(0,bmi,DIB_RGB_COLORS,&t,NULL,0);
		hMemDC=CreateCompatibleDC(GetDC(hWnd));
		SelectObject(hMemDC,hbmp);
		}
```

★ついにテスト期間です。うーん，つらい。（神奈川県川崎市・ミッチ20歳）……【**つぐ美：**で，結果はどうでした？何でもそうですけど，自分で納得できる結果だったなら，その苦労も吹き飛んじゃいますよね。**編：**苦労したことない人もいますけど。**影：**ぐははは。】

《リスト３》main.cppのプログラム・リスト

```
#include <windows.h>
#include "bcwin.h"
#include "bcbcc.h"

#define ID_TIMER        417

HINSTANCE       hInst;
TCHAR           szTitle[]="BC_BasicCharactical Class Demo";
TCHAR           szWindowClass[]="WndApp";
HDC                     mdc;
BCBC            *cs;
int mx,my;
int me;
int bx,by,bax,bay;
int t;
bool c[32][25];
int scine;
int score;
bool kl,kr,ks;

ATOM                            MyRegisterClass( HINSTANCE hInstance );
BOOL                            InitInstance( HINSTANCE, int );
LRESULT CALLBACK        WndProc( HWND, UINT, WPARAM, LPARAM );
void MainLoop(HWND hWnd);
void Game(void);
void draw(void);

int APIENTRY WinMain(HINSTANCE hInstance,HINSTANCE hPrevInstance,LPSTR lpCmdLine,int nCmdShow)
        {
        MSG msg;

        MyRegisterClass( hInstance );// グローバル ストリングを初期化します

        if( !InitInstance( hInstance, nCmdShow ) ) // アプリケーションの初期化を行います:
                {
                return FALSE;
                }

        while( GetMessage(&msg, NULL, 0, 0) )   // メイン メッセージ ループ:
                {
                TranslateMessage( &msg );
                DispatchMessage( &msg );
                }

        return msg.wParam;
        }

ATOM MyRegisterClass( HINSTANCE hInstance )
        {
        WNDCLASSEX wcex;

        wcex.cbSize = sizeof(WNDCLASSEX);

        wcex.style                      = CS_HREDRAW | CS_VREDRAW;
        wcex.lpfnWndProc        = (WNDPROC)WndProc;
        wcex.cbClsExtra         = 0;
        wcex.cbWndExtra         = 0;
        wcex.hInstance          = hInstance;
        wcex.hIcon                      = LoadIcon(hInstance, (LPCTSTR)IDI_APPLICATION);
        wcex.hCursor            = LoadCursor(NULL, IDC_ARROW);
        wcex.hbrBackground      = (HBRUSH)(COLOR_WINDOW+1);
        wcex.lpszMenuName       = (LPCSTR)0;
        wcex.lpszClassName      = szWindowClass;
        wcex.hIconSm            = LoadIcon(wcex.hInstance, (LPCTSTR)IDI_APPLICATION);

        return RegisterClassEx( &wcex );
        }

BOOL InitInstance( HINSTANCE hInstance, int nCmdShow )
        {
        HWND hWnd;

        hInst = hInstance; // グローバル変数にインスタンス ハンドルを保存します

        hWnd = CreateWindow(szWindowClass, szTitle,
WS_OVERLAPPED|WS_CAPTION|WS_SYSMENU|WS_MINIMIZEBOX,
                CW_USEDEFAULT, CW_USEDEFAULT, CW_USEDEFAULT, CW_USEDEFAULT, NULL, NULL,
hInstance, NULL);
        SetClientSize(hWnd,640,480);
        SetCenter(hWnd);

        if( !hWnd )
        {
                return FALSE;
        }

        ShowWindow( hWnd, nCmdShow );
        UpdateWindow( hWnd );

        return TRUE;
        }

LRESULT CALLBACK WndProc(HWND hWnd, UINT message, WPARAM wParam, LPARAM lParam)
        {
        PAINTSTRUCT ps;
        HDC hdc;

        switch( message )
                {
                case WM_KEYDOWN:
                        switch(wParam)
                                {
                                case VK_LEFT:   kl=true;break;
                                case VK_RIGHT:  kr=true;break;
                                case VK_SPACE:  ks=true;break;
                                }
                        break;
                case WM_KEYUP:
                        switch(wParam)
                                {
                                case VK_LEFT:   kl=false;break;
                                case VK_RIGHT:  kr=false;break;
                                case VK_SPACE:  ks=false;break;
                                }
                        break;
                case WM_PAINT:
                        hdc = BeginPaint (hWnd, &ps);
                        EndPaint( hWnd, &ps );
                        break;
                case WM_CREATE:
                        CreateCDC(hWnd,mdc,640,480);
                        cs=new BCBC(mdc,640,480);
                        //cs->font("ＭＳ 明朝");
                        cs->width(40,25);
                        scine=0;
                        SetTimer(hWnd,ID_TIMER,50,NULL);
                        break;
                case WM_DESTROY:
                        KillTimer(hWnd,ID_TIMER);
                        delete cs;
                        ReleaseDC(0,mdc);
                        PostQuitMessage(0);
                        break;
                case WM_TIMER:
                        MainLoop(hWnd);
                        break;
                default:
                        return DefWindowProc( hWnd, message, wParam, lParam );
                }
        return 0;
        }
```

★「kylix」をくわしく取り上げてください。（大阪市・**玉出のタマ35歳**）……**【影：**「kylix」とは，ボーランドのLinux用開発ツールで，X-Window環境でDelphiやVBのような快適なプログラ開発が可能になります。もちろん，今後ベーマガでも取り上げていきます。**】**

```
void MainLoop(HWND hWnd)
        {
        HDC hdc;
        int i,j;
        PatBlt(mdc,0,0,640,480,BLACKNESS);
        //この場合[cls]と同じようなもの。ようは黒く塗りつぶしてるだけ。
        //
        switch(scine)
                {
                case 0://タイトル
                        cs->locate(15,10);
                        cs->print("ブロック崩し");
                        me=3;
                        score=0;
                        t=0;
                        for(i=0;i<25;i++)
                                for(j=0;j<32;j++)
                                        c[j][i]=false;
                        for(i=3;i<8;i++)
                                for(j=2;j<30;j++)
                                        c[j][i]=true;
                        if(ks)
                                scine=1;
                        break;
                case 1:
                        mx=14;
                        my=24;
                        bax=1;
                        bay=1;
                        bx=15;
                        by=10;
                        t++;
                        draw();
                        if(t<10)
                                cs->print(15,10,"3");
                        else if(t<20)
                                cs->print(15,10,"2");
                        else if(t<30)
                                cs->print(15,10,"1");
                        else {t=0;scine=2;}
                        break;
                case 2:
                        Game();
                        break;
                case 3:
                        cs->color(255,255,255);
                        cs->locate(13,12);
                        cs->print("GAME OVER!!");
                        if(ks)
                                scine=0;
                        break;
                case 4:
                        cs->color(255,255,255);
                        cs->locate(13,12);
                        cs->print("Clear!!");
                        if(ks)
                                scine=0;
                        break;
                }
        //
        hdc=GetDC(hWnd);
        BitBlt(hdc,0,0,640,480,mdc,0,0,SRCCOPY);
        ReleaseDC(hWnd,hdc);
        return;
        }

void Game(void)
        {
        int i,j,k=0;
```

```
        if(kl && mx>1)
                mx--;
        if(kr && mx<25)
                mx++;
        if(ks)
                my=(my==24)?23:24;
        bx+=bax;
        by+=bay;
        bax=(bx==1 || bx==30)?bax*(-1):bax;
        bay=(by==1 || (by==(my-1) && bx>=mx && bx<mx+6))?bay*(-1):bay;
        if(by==30)
                {
                me--;
                scine=(me==0)?3:1;
                }
        if(by<10)
                if(c[bx][by])
                        {
                        score+=5;
                        c[bx][by]=false;
                        bay*=(-1);
                        if(c[bx+bax][by+bay])
                                bax*=(-1);
                        }
        for(i=3;i<8;i++)
                for(j=2;j<30;j++)
                        k+=(int)(c[j][i]);
        if(k==0)
                scine=4;
        //
        draw();
        cs->color(255,0,0);
        cs->print(bx,by,"@");
        return;
        }

void draw(void)
        {
        char tmp[10];
        int i,j;
        cs->color(0,0,255);
        cs->print(0,0,"################################");//#x32
        for(i=1;i<25;i++)
                {
                cs->color(i*5,i*5,255);
                cs->print(0,i,"#");
                cs->print(31,i,"#");
                }
        cs->color(255,0,255);
        cs->print(32,1, "////////");
        cs->print(32,3, "ブロック");
        cs->print(32,4, "  崩し  ");
        cs->print(32,6, "////////");
        cs->print(32,12,"スコア:" );
        wsprintf(tmp,"    %4d",score);
        cs->print(32,13,tmp);
        wsprintf(tmp,"残機数:%1d",me);
        cs->print(32,15,tmp);

        cs->color(255,255,0);
        for(i=0;i<25;i++)
                for(j=0;j<32;j++)
                        if(c[j][i])
                                cs->print(j,i,"#");
        cs->color(0,255,255);
        cs->print(mx,my,"======");
        return;
        }
```

★夏バテです。秋になったら「秋バテ」です。(神奈川県横浜市・**ケンタゴン18歳**)……【**影：**朝起きたら「朝バテ」で，昼寝すると「昼寝バテ」で，酒飲んで帰ると「バテバテ」ってか。**編：**影さんの1日の行動パターンですか。**影：**私の場合はバテ知らずだから。】

（超大作プログラム） リストが誌面に掲載できない作品です。
BMホームページ(http://www.basicmagazine.net/)からダウンロードしてお楽しみください。

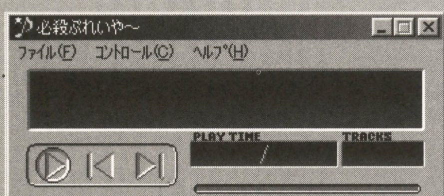
Windows(Visual C++ V6.0)用



音楽ファイルはおまかせ

必殺（ひっさつ）ぷれいや～

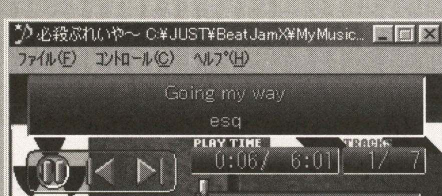
池添　貴之（19歳）



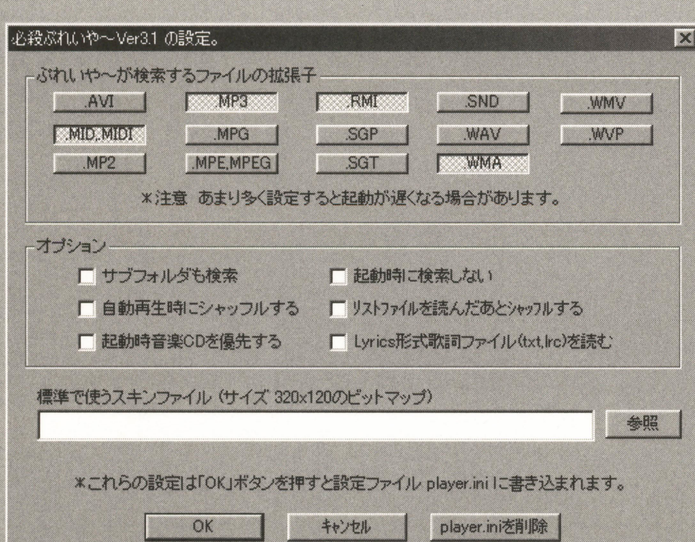
《写真１》数多くのフォーマットに対応



《写真２》MP3データのタイトルもバッチリ表示



《写真３》スキンファイルを作成すれば外見も思いのまま



《写真４》使用する環境に合わせた細かな設定も可能

●なんじゃこりゃ～!?●

必殺ぷれいや～は，かなり大量のメディアフォーマットに対応したシンプルかつ高性能なプレーヤです。

さらに，DirectMusic&DLSを使うことにより，サウンドボードに依存しない，そこらのプレイヤよりきれいなMIDIの再生が可能です。

最近のプレイヤによくある機能である，m3uリストファイルを使ったメディアファイルの連続再生，スキンの変更，歌詞表示もできます（細かいことについては，ダウンロードファイルに収録されているReadMe.htm参照）。

このツールは連続再生に特化しており，ゲーム開発者にはオススメです。

●通常の使いかた●

player.exeか，起動後，ウィンドウ上にメディアファイルをドラッグドロップするだけで再生できます（複数ファイルのドラッグ＆ドロップも可能です）。メニューから開くこともできます。

ウィンドウ左のボタンが灰色でない場合は「再生／停止」「前の曲へ」「次の曲へ」と同じ機能を持ったボタンを使用することができます。

ウィンドウ右下のシークバーは，クリックすると曲の再生位置を変えることができます。ドラッグはできません。

●自動再生機能●

player.exeの置いてあるフォルダに，

★（9月号73ページを見て）なるほど，それはいい考えだ…。ぜひお願いします。（神奈川県川崎市・**桜井道孝20歳**）……【**編**：テスターの使いかたということですか？検討しましょう。高校の物理部の取材に行ったときも，電子工作記事の要望をもらってきました。】

メディアファイルが存在している状態でそのまま実行，もしくはそれらのファイルの入ったフォルダをドラッグドロップして実行すると複数のファイルを読みこみ，順番に再生します（再生が終了するとつぎのファイルが再生され，すべて再生が終わるとまた一つめのファイルを再生します)。

player.exeの置いてあるフォルダに再生できるメディアファイルがないと，音楽CDの有無を確認し，なければ再生するファイルを選択できます。

この検索機能で列挙されるファイルを制限するには，メニューの「ファイル」から「ぷれいや〜の設定」で変更することができます。

ショートカット・キー

Ctrl+Oでファイル選択ダイアログを，スペース・キーで再生／停止切り替え，Alt+右キーor右ボタンクリックでつぎの曲へ，Alt+左キーで前の曲へ，左ダブルクリックで再生するファイルを選択できます。

ドラッグ＆ドロップについて

player.exeに対してドラッグドロップできるのは，

・単一のメディアファイルもしくはリストファイル
・複数のメディアファイル（一度に）
・単一のフォルダ

CHECKER FLAG

影：かなり完成された作りのメディアプレイヤだ。似たようなソフトはインターネット上などにも数多くあるので，アイデア面での面白味はないが，プログラム面での技術は非常に高度と言える内容だ。

Dr.D：高度な作りなだけに，リストの解析も初心者にとっては難しいじゃろう。ただ，メディアファイルの再生関連はすべて網羅しているリストとも言えるので，自分の手元に持っておけば，いざというときに参考になるリストになるハズじゃ。

編：音楽再生だけでなく，プログラム面でもベーマガ読者向けのプレイヤと言えるかもしれませんね。

《第１表》対応フォーマット

DirectX8，MediaPlayer7がインストールされている場合
スタンダードMIDI
一部のDirectMusicファイル(現在確認しているものは .sgt .sgp .wvp)
ウェーブサウンド
AU形式サウンド
Windows標準ビデオクリップ
ほとんどのMPEG(現在確認しているものは .mp3 .mpeg)
ほとんどのWindowsMediaフォーマット(現在確認しているものは .wma .wmv)
音楽ＣＤ
M3U形式再生リスト

※手元にメディアファイルがあまりないので，他にもまだ対応メディアファイルがあると思います。

必殺ぷれいや〜のウィンドウに対してドラッグドロップできるのは

・単一のメディアファイルもしくはリストファイル
・単一のフォルダ

です。

動作環境について

MP3などを再生する場合MediaPlayer6以上を，WMAやWMV等を再生する場合MediaPlayer7がインストールされている必要があります。他MPEGによってはcodecも必要です。

『必殺ぷれいや〜』には，DirectXのバージョンに左右されないよう，三つのバージョンを用意しています。

・**PlayerMCI.exe**：DirectXを使わないので，DirectMusicファイルの再生ができません。MIDIはMCIで再生します。
・**Player7.exe**：DirectX7を使用しますが，DirectMusicファイルがうまく再生されない場合があります。
・**Player.exe**：DirectX8を使用します。DirectX8がインストールされた環境では，このバージョンの使用を推奨します。

プログラムとワークスペースについて

DirectMusicは，よく変な仕様があるらしく，たとえば全角のファイル名を読むのによく失敗するので，対処方法として一時的にファイル名を半角ファイル名に変えたりしました。

また，MIDIがいつの間にかMCIで読み込まれていて，それに気づかずDirectMusicのバグだと思ってしまい，直すのに数日かかったこともありましたね。

バグを防止するために追加したコードが，また別のバグを出してしまうことだってあります。みなさんもデバッグには数週間ぐらいかけたほうがいいですよ〜（いや，マジで)。

改造のポイントとしては…スキンのレイアウトも変えられるようにする，リージョンを使ってウィンドウの形を変えたり…プレイヤの機能としては音量を変えられたりできるといいかも。ほか，歌詞が長いと省略されてしまうので，そのへんをなんとかする…いろいろありますね。

この必殺ぷれいや〜は，「必殺ぷれいや〜本体」「自作ライブラリ GCLIB（ビルドするとgclibStd.libを生成する)」の二つにわかれています。ライブラリについては，プレイヤに限らずVisualC++でWin32アプリケーションの作成に適しています。自由に使用してもかまいません。

program.txtにはプレイヤ本体のプログラムの流れと関数の簡単な説明について記述しています。

サンプルスキン２はDirectXのサンプルから，３，４の一部はネットで公開されているフリーのWeb素材を使わせてもらいました。

最後に宣伝。DirectMusicProducerはすごく使いやすいMIDIシーケンサです。DirectXのSDKに入っており，無料ですが高機能です（表示がすべて英語ですが…)。

★参考にしたもの

・DirectX7 実践プログラミング（工学社)
・DirectX8 実践プログラミング（工学社)
・マイコンBASICマガジン 2001年3月号
・DOS/V Magazine（SOFT BANK)

★　★　★

★「PS2 BASIC公開講座」に参加させていただきありがとうございました。電波新聞社の○川さんやアートディンクの斎藤さん，お世話になりました。あれからプログラム始めました。作品できたら投稿します。（栃木県宇都宮市・ヤマショウ41歳）……【編：はい。】

（超大作プログラム）　リストが誌面に掲載できない作品です。
BMホームページ(http://www.basicmagazine.net)からダウンロードしてお楽しみください。

Windows(Visual C++ V6.0)用



本格派の対戦型シューティング

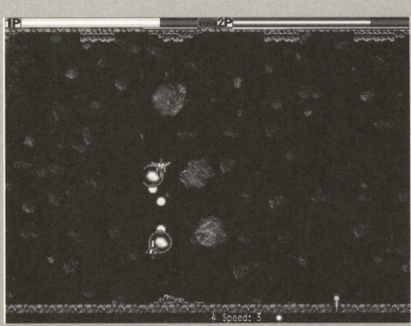
P.P.Year 7

（ピー ピー イヤー セブン）

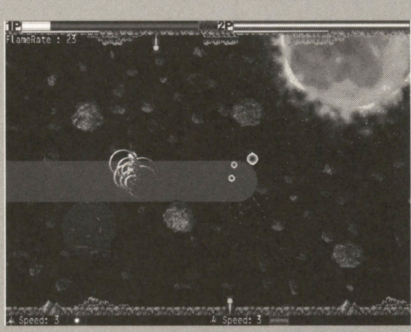
Gs（15歳）

内容

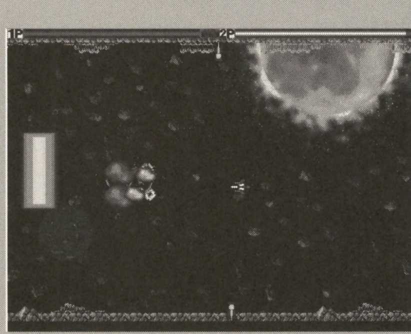
対戦型のシューティング・ゲーム？です。同レベルの装備を備える２体の機体が，様々な武器を駆使して相手を倒します。機体はそれぞれパワーを持ち，それが０になると負けです。



《写真１》２人で打ち合うタイプのシューティング



《写真２》派手な演出は一見の価値あり



《写真３》相手の機体を破壊すれば勝負あり

武器，アイテムなど，詳しいルールの詳細は，ダウンロードファイルに収録されているreadme.htmlをお読みください。

プログラムの仕組み

初のDirectX+VCプログラミングということで，プログラム的にはほとんど参考になる部分はないと思います。DirectXについては，すべて自作ライブラリで管理しています。その部分を見れば，ほんの少しは参考になるかもしれません。

このプログラムは，私がプログラムを始めたばかりのころ，VBからDirectXを使用せずに作成し，秒間0.5フレームという驚異的な数値を残した作品（PPYear3）をリメイクしたものです。

改造のアドバイスと苦労点

プログラムの本体はぐちゃぐちゃで，いじるのは無謀だと思いますが，ライブラリ部分はいくらでも変更可能だと思います。とくにGSDDraw.cppのライン命令は高速化の可能性がまだまだあります。

基本的にDirectXを使用して間もなくのころのライブラリなので，ほとんど利用価値はありません（今の最新版とはほとんど異なっています）。

《第１表》操作方法

キーボード	
プレイヤー１	
上	W（て）キー
下	S（と）キー
左	A（ち）キー
右	D（し）キー
ビーム	左Shiftキー
ミサイル	CapsLock（英数）キー
ボム	Q（た）キー
プレイヤー２	
上	↑キー
下	↓キー
左	←キー
右	→キー
ビーム	右Shiftキー
ミサイル	バックスラッシュ（ろ）キー
ボム	／（め）キー
ジョイスティック	
上	↑ボタン
下	↓ボタン
左	←ボタン
右	→ボタン
ビーム	1（A）ボタン
ミサイル	2（B）ボタン
ボム	3（X）（C）ボタン

SpecialThanks

BGM作成にMTRさん，またMIDI再生ライブラリにKBGMを利用しています。ありがとうございました。

CHECKER FLAG

つぐ美：画面写真からもわかるように，非常に完成された作品です。武器の種類などのゲームシステム面から演出面まで，どこを見てもしっかり作られているのが感じられます。

Dr.D：じゃが，やはりその分，リストはかなりの量じゃな。DirectX関連の記述も多くあるため，解析はチト面倒じゃが，このような完成された超大作作品がどのような感じで作られているかを，リスト面からも一度は見てみてはどうかな？

編：しいて難点を言うと，２人対戦のほうでプレイしたほうが面白いですが，プログラムの完成度という面から考えても，やはりCOM戦は付けてほしかったです。簡単な思考ルーチンのCOMでも，1Pモードがあると遊びやすいですからね。

★ドットは打っても打っても終わらないー。（福島市・南斗ライバ25歳）……【編：ドット絵描いてんの？何描いてんるのかなー，そんな終わらないほど大きなもの？Dr.D：ドットを打つじゃと？おっ，思い出した。鍼打ってくるな。予約しておった。一同：ぎゃふん。】

Java Script用



手軽な環境でもぐらたたき

モグラ叩き(たた)き in(イン) Java Script(ジャバ スクリプト)

MONOMON（31歳）

はじめに

単純なモグラ叩きです。初めてJavascriptを使って作ってみました。インターネットエクスプローラ(しかもVer5)しか正常に動かない(はず)ですが，まずまず遊べるものになったと思います。

遊びかた

STARTボタンをクリックすると始まります。モグラ(？)がつぎつぎと出てくるのでマウスでクリックして退治しましょう。モグラがめいっぱい出てるときに叩くと高得点です。また，泣いてるモグラや穴を叩くと減点です。１分経つとゲーム・オーバーです。

イメージについて

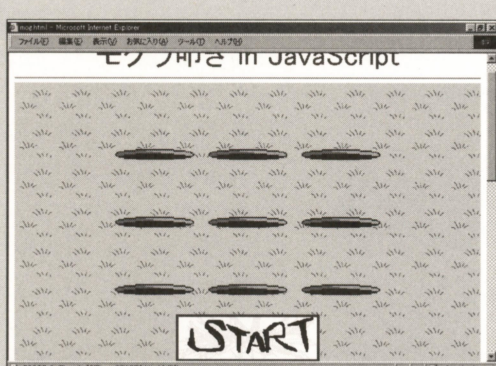
得点キャラ(ten*.png)，モグラキャラ(mogura*.png)は，周りを透過色にして保存してください。

ペイントブラシでBMPファイルを作成した後，Microsoft PhotoEditorでPNG形式に保存しなおすのですが，このツール上で透過色の指定ができます。

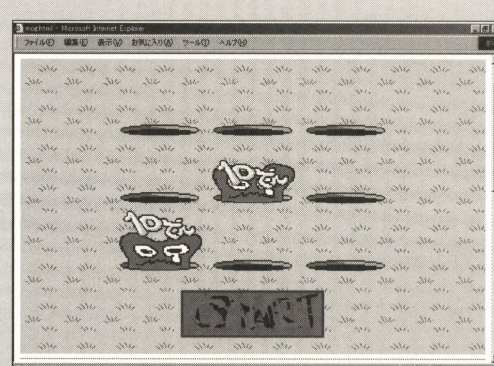
space.pngは，サイズの小さい全体が透過色のイメージです(座標合わせに使用しています)。

各イメージは，プログラム中でサイズを調整するので自由な大きさで作成してもいいのですが，一応サイズを示します。

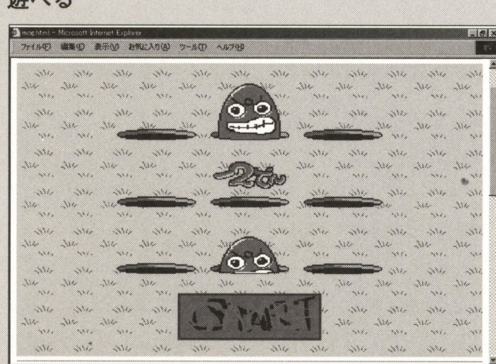
ten*.png：64×32
mogura*.png：64×64



《写真１》JavaScriptなのでブラウザがあれば遊べる



《写真２》ルールは単純，もぐらをたたくだけ



《写真３》間違ったところをたたくと減点



《写真４》ゲーム終盤はキツイ展開が待っている

kusa.png：64×64
space.png：2×2
start.png：256×128
game.png：256×128

サウンドについて

WAVEファイルを各自で用意してください（なくても動きますが，音が鳴りません)。用意するファイルは，以下の二つです。

・**poko.wav**:得点音(モグラを叩く)
・**boo.wav**:減点音(穴,泣きモグラをく)
ちなみに僕は「C:¥ProgramFiles¥Plus!¥Themes」に入ってたWAVEファイルをコピー＆リネームして使っています。
・**poko.wav**:ジャングル - メニュー ポップアップ.wav
・**boo.wav**:60 年代アメリカ - ビープ音.wav

楽しかったところ

JavaScriptは，HTMLタグの内容を直接参照・操作ができるのが使ってて楽しかったです。もともとBASIC言語が好きで，ソースの記述だけで動く手軽さが気に入ってるのですが（だからF-BASICもフォームをあまり使ってないのです)，この言語の場合，HTML文書でレイアウトを自由に作成し，それらを同じソース内のプログラムで動かせるので，自分にとってはある意味で「理想的な言語」のように感じています。

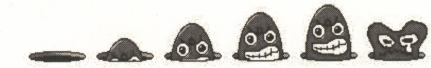
重たいし機種依存(ブラウザ依存)が大きすぎたりと何かと問題は多いですが，とりあえず楽しいってところがいいですね。

CHECKER FLAG

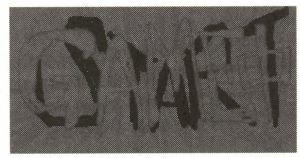
つぐ美：久々のJavaScriptプログラムです。内容は，良くも悪くも定番のもぐらたたきなので，新鮮味０ですが，ブラウザで手軽に遊べる環境の作品というのは評価できます。
Dr.D：全体的に動きに安定感がないのぉ。JavaScriptということもあるが，動作速度での工夫もほしいところじゃ。動作速度にかかわる面での工夫は，プログラム言語の仕様から見ると，チョット邪道と言われるような方法を取ったりもするが,それだけに腕の見せどころでもある。最近はハード側の速度が速いため，高速化へのこだわりを持った投稿プログラムも少ないが，ぜひともそういう投稿作品も待っておるゾ。



start.pngの画像


左からmogurao.png～mogura5.pngの画像


左からten0.png～ten6.pngの画像


game.png
の画像


kusa.png
の画像

苦労したところ

今回初めて扱う言語ということもあって，デバッグは苦労しました。

エラーが出てもエラーコードとかも出ないので，エラー行を特定していく作業が大変でした。しかも，エラーメッセージはすぐ消えるし…（これは自分が悪い。ステータスバーに得点を表示するから)。ネットスケープにも対応したかったのですが，持ってなくて確認できないのでやめました。

バグ情報，改造など

ウィンドウサイズに応じてキャラサイズを変更するようになってるのですが，小さくしすぎると表示がずれるみたいです（なんでかな？)。

改造については，他のキャラやアイテムを出したり，出てくる場所を自然に散らかしたり，クッキーを使ってハイスコアを保存したり，見にくいスコアとタイムを画面中に出したりといろいろありますね。

実際にネット上にアップすることを考えるなら，ネットスケープに対応したり，ハイスコア投稿用のフォームをつけたり(集計用のCGIがいるけど…)，戻るボタンをつけたりとかですね。

いろいろ試そう

今回，いろいろ苦労をしながら作ったのですが，その過程のバージョンでいろんな作りかたが出てきました。

たとえば，音を出す部分も，最初はフレームを使って<embed>タグでサウンドを貼り付けたHTMLを見えないように配置し，location.reloadメソッドを使って鳴らしてましたが，<bgsound>タグを直接操作できることを知ってこの形になおしました。

また，得点キャラを出す部分も，最初はイメージを使わずテキストで表示させてましたが，重たいので変更しました。

そして得点キャラが出現中でもモグラをクリックできるように，はじめは得点キャラの<a>タグを書き換えてたのですが，効率が悪いので上からスイッチキャラをかぶせる今のシステムにしました。

このスイッチも，はじめは透明イメージを並べてたのですが，プレイ中の操作ミスで範囲指定をしてしまった場合，最上位の透明イメージが真っ青になりモグラや穴が見えなくなるので，表組みそのものをスイッチにする方法をとりました。

このように同じ効果を出すにも実現方法がいろいろあり，アルゴリズム的に方法論がいっぱいあるだけのと違って，同じような結果を生むコマンドが多数あるというのもこの言語の特色なのかなと思いました。

不可能だと思っても何かと逃げ道がいっぱい用意されてる言語だと思うので，あきらめずにいろいろ試せば多言語に負けないくらい，いろんなことができそうです。

★新ゲーム機「Xbox」，発売が楽しみです。私は今まで，すべてのゲーム機を触ってきました。でも，実際に持っている（持っていた）のは，ファミコン，C1，スーパーファミコン，PS2だけです。あとは借り物。（東京都あきる野市・柴田22歳）……【影：で，Xboxは？】

『モグラ叩きin Java Script』のプログラム・リスト

```
<html>
<head>
<title>モグラ叩き</title>
<script language="JavaScript">
<!--

//#### 初期設定 ########
MF = new Array(8);      // モグラフラグ
MT = new Array(8);      // モグラタイマー
MS = new Array(8);      // モグラ速度
MW = new Array(8);      // モグラ待機時間
M  = new Array(8);      // モグラ移動方向(1:出る,-1:戻る)

MSC = new Array(-2,2,4,6,10,-10);      // 各モグラの得点

MC = 0;                 // モグラ出現タイミング
MCT = 0;                // モグラ出現タイマー

tenNo = -1;             // 得点キャラ出現数
tenTM = new Array(2);   // 得点キャラタイマー
tenSp = 0;              // 得点キャラ速度

GF = false;             // ゲームフラグ(true:ゲーム中, false:ゲーム終了)
GT = 0;                 // プレイ時間
SC = 0;                 // 得点
TMID = 0;               // タイマーID
TM = 50;                // 割り込み時間(ミリ秒)

//#### 画面初期化 ########
function init() {
  fW = document.all.ana.style.pixelWidth;      // フィールド幅
  fH = document.all.ana.style.pixelHeight;     // フィールド高さ
  fL = document.all.ana.style.pixelLeft;       // フィールドX
  fT = document.all.ana.style.pixelTop;        // フィールドY

  img_gam = new Image;          // イメージのプレロード
  img_gam.src = "game.png"
  score();
}

//#### ゲーム開始：スタートボタンクリック ########
function start() {
  if (GF) return;

  document.sw.src = "game.png";
  for (var i=0; i<9; i++) { MF[i] = 0; mogurawrite(i); }
  for (i=tenNo; i>=0; i--) document.all["ten"+i].style.visibility = "hidden";
  tenNo = -1;

  MC = 1500;  MCT = 500;
  SC = 0   ;  GT = 60 * 1000;
  GF = true;

  score();
  scroll(0,document.all.field.offsetTop - 8);

  TMID = setTimeout('main()',TM);      // タイマースタート
}

//#### メイン：ゲームループ ########
function main() {
  var i,j;
  if ( (MCT -= TM) <= 0 ) {     // モグラ出現
    i = parseInt(Math.random()*9);      // 穴の選択
    if (MF[i] == 0) {                   // その穴にモグラがいなければ出現
      MF[i] = 1;
      j = parseInt(MC/10);
      MS[i] = parseInt(Math.random()*3+1)*j;
      MW[i] = parseInt(Math.random()*3+2)*j;
      MT[i] = MS[i];
      M[i]  = 1;

      if ( (MC -= 20) <= 150 ) MC = 150;
      MCT = MC - parseInt(Math.random()*2)*100;
    }
  }

  for (i=0; i<9; i++) {          // 各モグラの動き
    if (MF[i] == 0) continue;
    if (MF[i] == 4) {            // モグラが完全に出きってるとき
      if ( (MW[i] -= TM) <= 0 ) {
        M[i] = -1;
        MF[i] += M[i];
        mogurawrite(i);
      }
    } else {                     // モグラが出る途中、引っ込む途中、やられ中
      if ( (MT[i] -= TM) <= 0 ) {
        MT[i] = MS[i];
        MF[i] += M[i];
        mogurawrite(i);
      }
    }
  }

  score();

  for (i=0; i<=tenNo; i++) {    // 得点キャラの移動
    tenTM[i] += TM;
    document.all["ten"+i].style.pixelTop = document.all["ten"+i].style.pixelTop+tenSp*TM;
  }

  if (tenNo > -1 && tenTM[tenNo] > 1000) {      // 得点キャラの消去
    document.all["ten"+(tenNo--)].style.visibility = "hidden";
  }

  clearTimeout(TMID);

  if ( (GT -= TM) <= 0 ) {      // ゲーム終了
    document.sw.src = "start.png";
    GF = false;  return;
  }
  TMID = setTimeout('main()',TM);

}

//#### モグラアニメ：絵の切り替え(n:モグラの穴の位置) ########
function mogurawrite(n) { document["mog"+n].src = "mogura"+MF[n]+".png"; }

//#### モグラを叩く：モグライメージクリック ########
function moguraclick(n) {
  if (!GF) return;
  SC += MSC[ MF[n] ];
  tokuten(n);
  if (MF[n] > 0 && MF[n] < 5) { // モグラ出現中(加点&やられにする)
    MF[n] = 5;
    M[n]  = -5;
    MT[n] = 1000;
    POKO.src = "poko.wav"
  } else {                      // モグラいない(減点&穴に戻す)
    MF[n] = 0;
    BOO.src = "boo.wav"
  }
  score();
  mogurawrite(n);
}

//#### 得点表示キャラ登場 ########
function tokuten(n) {
  if (tenNo == 2) tenNo--;      // 得点キャラが2(MAX)なら1つ減らす
  for (var i=tenNo; i>=0; i--) {        // 各得点キャラを後ろへずらす
    tenTM[i+1] = tenTM[i];
    document["ten"+(i+1)+"chr"].src = document["ten"+i+"chr"].src;
```

★ついに携帯電話を持ってしまいました。ふん，負け犬と呼ぶなら呼ぶがいいさ！！（岩手県釜石市・**鉄人坂井23歳**）……【**つぐ美：**影さんだって，最初は「絶対持たない」なんて言ってたのに，今じゃ，肩身離さず持ち歩いてます。**影：**電話かかってこないかなー。】

```
    document.all["ten"+(i+1)].style.pixelLeft = document.all["ten"+i].style.pixelLeft;
    document.all["ten"+(i+1)].style.pixelTop  = document.all["ten"+i].style.pixelTop;
  }
tenTM[0] = 0;                    // で、0番に新規キャラを作成
document.ten0chr.src = "ten"+MF[n]+".png";
document.all.ten0.style.pixelLeft = fL + fW/3*(n%3+0.5) - document.all.ten0.offsetWidth/2;
document.all.ten0.style.pixelTop=fT+fH/3*(parseInt(n/3)+1)-document.all.ten0.offsetHeight;
tenSp = -(fH/3-document.all.ten0.offsetHeight)/1000;
document.all["ten"+(++tenNo)].style.visibility = "visible";
}

//#### サイズ変更時のサイズ確認 ########
function sizechange() {
  var oL=fL, oW=fW, oT=fT, oH=fH        // 古い情報を保存
  fL = document.all.ana.style.pixelLeft          // 新X
  fT = document.all.ana.style.pixelTop           // 新Y
  fW = document.all.ana.style.pixelWidth         // 新幅
  fH = document.all.ana.style.pixelHeight        // 新高さ
for (var i=0; i<=tenNo; i++) {  // 各得点キャラの座標を再設定
document.all["ten"+i].style.pixelLeft=((document.all["ten"+i].style.pixelLeft-oL)/oW)*fW+fL;
document.all["ten"+i].style.pixelTop=((document.all["ten"+i].style.pixelTop-oT)/oH)*fH+fT;
}
  tenSp = -(fH/3-document.all.ten0.offsetHeight)/1000;          // スピード再設定
  scroll(0,document.all.field.offsetTop - 8);                   // 再位置あわせ
}

//#### スコア表示 ########
function score() {
  status = " SCORE: " + SC + "  Time: " + ( Math.floor(GT/100) / 10 ) + ( (GF)?"  ゲーム中  頑
張ってね":"  終了      STARTをクリックしてね" );
}

//-->
</script>
</head>

<body bgcolor = "#ffffff" text = "#000000" onLoad = "init();" onUnload = "clearTimeout(TMID);"
onresize = "sizechange();">
<bgsound src="poko.wav" id="POKO">
<bgsound src="boo.wav"  id="BOO">
<center>

<!-- タイトル記述 --------------------------------->
<font size="+8" color="#0000ff">モグラ叩き in JavaScript</font><br>

<!-- ゲームフィールド描画 -------------------------->
<hr>

<div id="field" style="background-image: url(kusa.png)">
  <img src="space.png" width="10%" height="5%"><br>
  <table border="0" width="60%" height="70%" cellspacing="0" cellpadding="0"><tr><td>
    <div id="ana" style="position:absolute;width:100%;height:100%">
      <table border="0" width="100%" height="100%" cellspacing="0" cellpadding="0">
        <tr>
        <td><img name="mog0" src="mogura0.png" width="100%" height="100%" border="0"></td>
        <td><img name="mog1" src="mogura0.png" width="100%" height="100%" border="0"></td>
        <td><img name="mog2" src="mogura0.png" width="100%" height="100%" border="0"></td>
        </tr><tr>
        <td><img name="mog3" src="mogura0.png" width="100%" height="100%" border="0"></td>
        <td><img name="mog4" src="mogura0.png" width="100%" height="100%" border="0"></td>
        <td><img name="mog5" src="mogura0.png" width="100%" height="100%" border="0"></td>
        </tr><tr>
        <td><img name="mog6" src="mogura0.png" width="100%" height="100%" border="0"></td>
        <td><img name="mog7" src="mogura0.png" width="100%" height="100%" border="0"></td>
        <td><img name="mog8" src="mogura0.png" width="100%" height="100%" border="0"></td>
        </tr>
      </table>

<!-- 得点キャラ(スプライト)を作成 ------------------>

<div id="ten0" style="position:absolute; z-index:3; visibility:hidden; width:30%; height:15%">
<img src="ten0.png" name="ten0chr" width="100%" height="100%" border="0"></div>
<div id="ten1" style="position:absolute; z-index:2; visibility:hidden; width:30%; height:15%">
<img src="ten0.png" name="ten1chr" width="100%" height="100%" border="0"></div>
<div id="ten2" style="position:absolute; z-index:1; visibility:hidden; width:30%; height:15%">
<img src="ten0.png" name="ten2chr" width="100%" height="100%" border="0"></div>
    </div>

<!-- 透明スイッチをセットする(表のセルそのものがスイッチ) -->

    <div style="position:relative; left:0; top:0 z-index:10">
      <table border="0" width="100%" height="100%" cellspacing="0" cellpadding="0">
        <tr>
        <a onmousedown="moguraclick(0);"><td></td></a>
        <a onmousedown="moguraclick(1);"><td></td></a>
        <a onmousedown="moguraclick(2);"><td></td></a>
        </tr><tr>
        <a onmousedown="moguraclick(3);"><td></td></a>
        <a onmousedown="moguraclick(4);"><td></td></a>
        <a onmousedown="moguraclick(5);"><td></td></a>
        </tr><tr>
        <a onmousedown="moguraclick(6);"><td></td></a>
        <a onmousedown="moguraclick(7);"><td></td></a>
        <a onmousedown="moguraclick(8);"><td></td></a>
        </tr>
      </table>
    </div>
  </td></tr></table>

  <img src="space.png" width="10%" height="5%"><br>
  <a href="JavaScript:start();" onmouseover="score(); return true;">
    <img name="sw" src="start.png" width="30%" height="15%" border="4"></a><br>
  <img src="space.png" width="10%" height="5%">
</div>


<!-- マニュアル記述 --------------------------------->
<hr>
<br>
<font size="+5">遊び方</font><br>
<br>
モグラ(?)が穴から出てくるのでマウスでクリックして懲らしめてください。<br>
モグラ(?)の頭が完全に出てる時に叩けば高得点です。<br>
でも穴を叩いてしまうと、せっかく生えそろった芝生を荒らしてしまうので減点です。<br>
また、泣いて改心したモグラ(?)に追い討ちをかけると、動物(?)愛護協会からお叱りを受けて大減点で
す。<br>
1分間もったら勤務終了です。画面下のスコアがあなたの能力給です。<br>
<br>
<br>
<font size="+3" color="#000000">
得点表<br>
<img src="mogura4.png" width="64" height="64">………<img src="ten4.png" width="64"
height="32"><br>
<img src="mogura3.png" width="64" height="64">………<img src="ten3.png" width="64"
height="32"><br>
<img src="mogura2.png" width="64" height="64">………<img src="ten2.png" width="64"
height="32"><br>
<img src="mogura1.png" width="64" height="64">………<img src="ten1.png" width="64"
height="32"><br>
</font>
<br>
<font size="+3" color="#ff0000">
<img src="mogura0.png" width="64" height="64">………<img src="ten0.png" width="64"
height="32"><br>
<img src="mogura5.png" width="64" height="64">………<img src="ten5.png" width="64"
height="32"><br>
</font>

</center>
</body>
</html>
```


★私，惚れました。ともえちゃんと影さんに。(静岡県磐田市・**名前は絶対出さないで15歳**)……【**つぐ美**：あら，ほほえましいわ。ともえちゃんって，学校のお友達ネ。ところで，影さんって？**影**：おいおい，ここここ。**つぐ美**：ああ，そっか。**影**：そりゃないぜ。】



牛食えキャラに牛丼一丁

牛丼食えスト外伝　牛丼道

（ぎゅうどんくえすとがいでん　ぎゅうどんどう）

ヅラＣｈｕ（18歳）

はじめに

久しぶりのプログラム投稿です。503iを買ったので，iアプリで『牛食え』のミニゲームを作ってみました。iアプリは容量の制約があるところが昔のBASICぽくて燃えます（笑）。それに容量が少ない分お手軽に組めるところがナイス。

あらすじ

吉○家に対抗して並盛を29G＄（ギュードル）に値下げしたので，主人公君の牛丼屋は今日も大忙しなのでした。って，来るの身内ばっかりじゃん。

にゅうりょく

開発環境は，Java2SDK1.3とJ2ME WirelessSDKforDoJa。普通はこれだけで十分なのですが，J2MEに付属のKToolbarが使いにくさ抜群なので，開発には西村誠一氏作の『iアプリ開発Tool』（フリーソフト）を使わせていただきました。ビルドのときにコンソールを開く必要がないし，jamファイルやDLページも自動生成してくれるので楽チン。

エミュレータはi-JADE Liteを使ってますが，絵文字が表示できないため，絵文字の確認に仕方なくDoJa付属のエミュレータを併用してます。7月号iアプリ講座のように実機のデータを入れないと画面サイズとかがおかしいので注意。

そんなわけでやっとこさプログラム入力。gyudon.javaをお使いの環境で入力してください。画像は牛丼食えスト公式サイトの素材を改変して使わせていただきました。それぞれ《第１表》に示すファイル名，画像サイズで保存します。

あとはご使用の環境でコンパイル，プリベリファイ，jar圧縮を行い，jamファイルを作成すれば完成です。

《第１表》画像ファイルについて

画　像	概要	ファイル名	幅	高
	主人公	shujinko.gif	20	40
	カウベル	cowbell.gif	20	20
	ＪＡ	ja.gif	20	20
	ケンタ	kenta.gif	20	20
	マイ姉	mai.gif	20	20
	牛丼大王	daioh.gif	20	20
	ゴーレム	gorem.gif	20	20
	フキダシ	name.gif	20	20
牛食え	牛食えロゴ	logo.gif	47	15

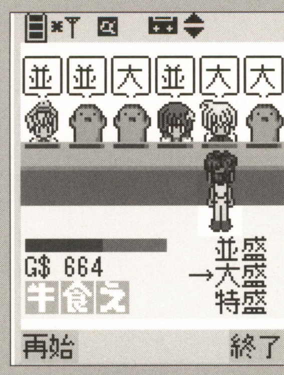
あそびかた

主人公君は左右キーで左右に移動，上下キーで出す商品を選択し，決定キーでお客に商品を出します。テンキーの1，2，3でそれぞれ並盛，大盛，特盛にカーソルを合わせることもできます。

お店にはつぎつぎとお客がやってくるので，フキダシをよく見て適切な商品を出してあげると，並盛なら29G＄，大盛なら39G＄，特盛なら49G＄もらえ，また店の評判ゲージが上がっていきます。評判が上がるとお客も増え，つまりは難易度が上がります。売上げの増加でも難



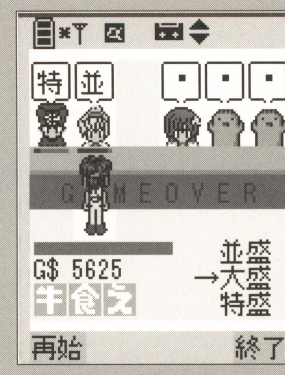
《写真１》牛食えキャラへ牛丼を出す



《写真２》キャラクターの要求を素早くこなせ



《写真３》キャラクターによって待ち時間などが異なる



《写真４》評判ゲージが底をつくとゲーム終了

★うきうき気分でこの前，Windows98からMeにアップグレードしたら，むちゃくちゃ調子が悪い！98に戻したら，よけいおかしくなって，結局，再インストールしました…（泣）。（京都市・Essa14歳）……【影：わかるわかる。こんなことの繰り返しが勉強になる。】

易度は上がります。

間違った商品を出してしまった場合，お客は怒って帰ってしまい，店の評判が下がります。また，お客の前にあるゲージがだんだん減少していきますが，これはそれぞれのお客の待ち時間を表していて，これを越えても同様です。評判ゲージが0になるとゲーム・オーバーです。

きゃらくたー

ベーマガ読者なら大体わかると思うので，細かいキャラ紹介は略で。同じキャラが2人以上同時に出てくるのは仕様なので気にしないように。

・**カウベル・ＪＡ**：至って普通のお客。

・**ケンタ**：短気なので他の客に比べて待ち時間が短い。

・**マイ**：来店時にゴーレムを呼び出して一気に空いている席をすべて埋めます。ただし，ゴーレムは待ち時間が他の客より長く，評判の減少も他の客より少ない。

・**牛丼大王**：怒って帰るとき，評判が他の客の2倍下がります。食い逃げはありません（笑）。

ぷろぐらむ

とくに込み入ったことはしてませんが，ベーマガのiアプリ講座とは違い，ソースファイルを一つにまとめてあります。自分にしてはコメントを結構つけたほうなので，頑張って読めばプログラムを解読できると思います。

機種依存について。まず表示に関しては，Paintメソッド内で画面サイズ，フォントサイズの調整を行っているので，多分どの機種でも同じような画面で実行できるはずです。D503i，N503i（以上実機），F503i，P503i（以上i-JADE Lite（絵文字は・と表示される）），DoJaエミュレータでの動作を確認済み。ただしカラーテーブルの都合で，主人公君の周りだけ浮いて見える場合があります。あとN503iではキー連打の反応が悪かったです。また速度のほうは解消しようがないので，F503iとP503iでの動作速度も怪しいかも。なお，SO503iについては未確認です。

かいぞう

簡単なところでは，難易度を変更したり，新キャラを出してみたりしてはどうかと。自分自身の課題としては，CGIと連動してネット上でスコアランキングを実現したいです。

CHECKER FLAG

影：牛食えキャラが多数出現するゲーム。Web上で展開中の牛食えではあるけど，まだまだベーマガ読者での人気は高い。最近では，こういった投稿プログラムでの素材にキャラクター画像がよく使用されているなぁ。

つぐ美：単純な内容ですが，ついついハマってしまうゲーム。そういう意味でもiアプリ向けの作品と言えるわね。

Dr.D：ゲームシステムの肝となる部分は，牛丼の量の判断だけなので，if文の組み合わせでおおまかな流れが実現できる。容量に制限のあるiアプリで多くの画像ファイルを使用するには，こういうゲームシステムそのものの考慮も重要になってくるゾ。

《リスト１》gyudon.javaのプログラム・リスト

```
//牛丼道
//(c)2001 ZR2U

import com.nttdocomo.ui.*;
import com.nttdocomo.util.*;
import com.nttdocomo.io.*;
import java.util.*;
import java.io.*;

public class gyudon extends IApplication {

        public void start(){
                myCanvas c=new myCanvas();
                Display.setCurrent(c);
        }
}

class myCanvas extends Canvas implements Runnable{

        Thread myThread;

        Image imgMy;
        Image imgGs1;
        Image imgGs2;
        Image imgGs3;
        Image imgGs4;
        Image imgGs5;
        Image imgGs6;
        Image imgName;
        Image imgLogo;

        int myX=2;      //主人公君の座標
        int myItem=0;   //出せる商品
        int myMoney;    //売り上げ
        int myPop;      //評判

        int[] gs=new int[6];    //お客
        int[] gsStatus=new int[6];      //お客のステータス
        int[] gsTime=new int[6];        //お客の待ち時間
        int[] gsCount=new int[6];       //お客の待ち時間の減り具合
        int[] gsDown=new int[6];        //お客が怒った時の評判の下がり具合
        String[] gsName={"並","大","特","¥uE6F0","¥uE6F1"};     //吹き出し表示データ

        Random rnd=new Random();

        public Image getImg(String sName){      //イメージ取得関数
                MediaImage mi=MediaManager.getImage(sName);
                try{
                        mi.use();
                }catch(ConnectionException e){}
                return(mi.getImage());
        }

        public void subInit(){  //ゲーム初期化関数
                myPop=50;
                myMoney=0;

                int i;
                for(i=0;i<6;i++){
                        gs[i]=0;
                        gsStatus[i]=3;
                }
        }

        public myCanvas(){      //コンストラクタ
                setSoftLabel(Frame.SOFT_KEY_1,"再始");
                setSoftLabel(Frame.SOFT_KEY_2,"終了");

                imgMy=getImg("resource:///shujinko.gif");       //主人公君
```


★テレビを見ながら，ポケーっとインターネットすることが好きなのですが…。（愛知県刈谷市・せゆ26歳）……【影：テレビつけながら（音消して），ステレオで音楽聴いて，インターネットしつつ，小説を読む。これが私の休日の過ごしかた。一同：ウッソだー。】

```
		imgGs1=getImg("resource:///cowbell.gif");		//カウベル
		imgGs2=getImg("resource:///ja.gif");	//JA
		imgGs3=getImg("resource:///kenta.gif"); //ケンタ
		imgGs4=getImg("resource:///mai.gif");	//マイ姉
		imgGs5=getImg("resource:///daioh.gif"); //大王
		imgGs6=getImg("resource:///gorem.gif"); //ゴーレム
		imgName=getImg("resource:///name.gif"); //吹き出し
		imgLogo=getImg("resource:///logo.gif"); //牛食えロゴ

		myThread=new Thread(this);		//スレッド作成・開始
		myThread.start();
		subInit();
	}

	public void paint(Graphics g){	//ペイントメソッド
		int i;	//ループ用

		g.lock();		//画面のロック

		//画面サイズ・配置の調整
		g.setOrigin((getWidth()-120)/2,(getHeight()-120)/2);
		g.clearRect(0,0,120,120);

		//フォントサイズの調整
		Font f=Font.getDefaultFont();
		if(f.getAscent()>13){
			f=Font.getFont(f.SIZE_SMALL);
		}
		g.setFont(f);

		g.setColor(g.getColorOfRGB(238,170,0)); //カウンター
		g.fillRect(0,40,120,11);
		g.setColor(g.getColorOfRGB(221,153,0));
		g.fillRect(0,51,120,2);
		g.setColor(g.getColorOfRGB(170,119,0));
		g.fillRect(0,53,120,16);

		g.drawImage(imgLogo,2,103);		//牛食えロゴ

		g.setColor(g.getColorOfName(g.BLACK));	//文字
		g.drawString("並盛",88,93);
		g.drawString("大盛",88,105);
		g.drawString("特盛",88,117);
		g.drawString("G$",2,101);

		g.drawString(Integer.toString(myMoney),20,101); //売り上げ
		g.drawString("→",76,myItem*12+93);		//商品選択カーソル

		g.drawImage(imgMy,myX*20,43);	//主人公君

		g.setColor(g.getColorOfName(g.RED));		//評判ゲージ
		g.fillRect(2,84,64,6);
		g.setColor(g.getColorOfName(g.BLUE));
		g.fillRect(2,84,myPop*64/100,6);

		for(i=0;i<6;i++){		//お客
			if(gs[i]!=0){
				switch(gs[i]){
				    case 1:g.drawImage(imgGs1,i*20,20);break;//カウベル
				    case 2:g.drawImage(imgGs2,i*20,20);break;//JA
				    case 3:g.drawImage(imgGs3,i*20,20);break;//ケンタ
				    case 4:g.drawImage(imgGs4,i*20,20);break;//マイ姉
				    case 5:g.drawImage(imgGs5,i*20,20);break;//大王
				    case 6:g.drawImage(imgGs6,i*20,20);break;//ゴーレム
				}
				g.drawImage(imgName,i*20,0);	//吹き出し
				g.setColor(g.getColorOfName(g.BLACK));
				g.drawString(gsName[gsStatus[i]],i*20+4,15);
				if(gsStatus[i]<3){		//商品待ちの場合
				    g.setColor(g.getColorOfName(g.RED));//待ち時間ゲージ
				    g.fillRect(i*20+2,41,16,2);
				    g.setColor(g.getColorOfName(g.BLUE));
				    g.fillRect(i*20+2,41,gsTime[i]*16/1000,2);
				}
			}
		}

		if(myPop<=0){	//ゲームオーバー時の文字描画
			g.setColor(g.getColorOfName(g.BLUE));
			g.drawString("G A M E O V E R",15,66);
		}

		g.unlock(false);		//画面のロック解除
	}

	public void run(){		//スレッド実行
		int iKey;		//キー情報を格納
		int iLeft=0,iRight=0,iUp=0,iDown=0;	//連続移動防止用
		int iWait=0;	//ゲームオーバー時のウェイト
		int i,j;		//ループ用

		for(;;){		//無限ループ

			try{	//ウェイト(50ms)
				Thread.sleep(50);
			}catch(InterruptedException ignore){}

			iKey=getKeypadState();	//キー情報の取得

			if(myPop>0){	//非ゲームオーバー時

			    if(iKey==(1<<Display.KEY_LEFT)&&(myX>0)){//主人公君の移動（左）
					if(iLeft==0){
						myX--;
						iLeft=1;
					}
			    }
			    else{
				iLeft=0;
			    }
			    if(iKey==(1<<Display.KEY_RIGHT)&&(myX<5)){//主人公君の移動（右）
				if(iRight==0){
					myX++;
					iRight=1;
				}
			    }
				else{
					iRight=0;
				}
				if(iKey==(1<<Display.KEY_UP)){	//出す商品の変更（上）
					if(iUp==0){
						myItem--;
						if(myItem<0){
							myItem=2;
						}
						iUp=1;
					}
				}
				else{
					iUp=0;
				}
				if(iKey==(1<<Display.KEY_DOWN)){//出す商品の変更（下）
					if(iDown==0){
						myItem++;
						if(myItem>2){
							myItem=0;
						}
						iDown=1;
					}
				}
```

★9月号掲載のプログラムで，フラグのことをフラッグにしてしまった。それじゃ旗だろう。（長野市・**宮尾　渉13歳**）……【**編**：設定など状態を記録している部分やファイルを旗に見立てて「flag」と呼ぶから，まちがってないよ。カタカナ表記はフラグが多いけど。】

```
else{
        iDown=0;
}
if(iKey==(1<<Display.KEY_1)){   //出す商品の変更（並盛）
        myItem=0;
}
if(iKey==(1<<Display.KEY_2)){   //出す商品の変更（大盛）
        myItem=1;
}
if(iKey==(1<<Display.KEY_3)){   //出す商品の変更（特盛）
        myItem=2;
}

//商品を出す
if((iKey==(1<<Display.KEY_SELECT))&&(gsStatus[myX]<3)){
        if(myItem==gsStatus[myX]){//希望商品だった場合
                gsStatus[myX]=3;
                myMoney+=10*myItem+29;
                myPop++;
                if(myPop>100){
                        myPop=100;
                }
        }
        else{   //間違っていた場合
                gsStatus[myX]=4;
                myPop-=gsDown[myX];
                if(myPop<0){
                        myPop=0;
                }
        }
        gsTime[myX]=20;
        gsCount[myX]=1;
}

//お客の処理
for(i=0;i<6;i++){
        if(gs[i]!=0){
                gsTime[i]-=gsCount[i];
                if(gsTime[i]<=0){//待ち時間切れの判定
                if(gsStatus[i]<3){//商品待ちの場合怒って帰る
                        gsStatus[i]=4;
                        myPop-=gsDown[i];
                        if(myPop<0){
                                myPop=0;
                        }
                        gsTime[i]=20;
                        gsCount[i]=1;
                }
                    else{//表情表示の場合お客を消去
                        if(gsTime[i]<=0){
                                gs[i]=0;
                        }
                    }
                }
        }
}

//新しいお客の登場
if(Math.abs(rnd.nextInt()%2000)<myPop+100){
        i=Math.abs(rnd.nextInt()%6);
        if(gs[i]==0){
                gs[i]=Math.abs(rnd.nextInt()%5)+1;
                gsStatus[i]=Math.abs(rnd.nextInt()%3);
                gsTime[i]=1000;
                gsCount[i]=myMoney/500+10;
                gsDown[i]=5;
                switch(gs[i]){
                case 3://ケンタ
                        gsCount[i]=myMoney/500+20;
                        break;
                case 4://マイ姉
                    for(j=0;j<6;j++){
                    if(gs[j]==0){//ゴーレム生成
                      gs[j]=6;
                      gsStatus[j]=Math.abs(rnd.nextInt()%3);
                      gsTime[j]=1000;
                      gsCount[j]=myMoney/500+5;
                      gsDown[j]=3;
                    }
                    }break;
                case 5:gsDown[i]=10;break;//大王
                }
        }
}

//再始判定
if(iKey==(1<<Display.KEY_SOFT1)){
        myPop=0;
}

//ゲームオーバー判定
if(myPop<=0){
        iWait=100;
}
}
else{   //ゲームオーバー時
        iWait--;
        if(iWait<=0){
                subInit();
        }
}

//終了判定
if(iKey==(1<<Display.KEY_SOFT2)){
        IApplication.getCurrentApp().terminate();
}

repaint();      //再描画
}
}
}
```

《リスト２》gyudon.jamのリスト

```
AppName = 牛丼道
PackageURL = gyudon.jar
AppSize = 5748
AppClass = gyudon
LastModified = Thu, 14 Jun 2001 12:51:30
AppVer = 1.2
KvmVer = CLDC-1.0
```

《リスト３》HTMLファイルの記述例

```
<html>
<HEAD>
<META http-equiv="Content-Type" content="text/html; charset=Shift_JIS">
<TITLE>gyudon</TITLE>
</HEAD>
<body>


<OBJECT declare id="gyudon" data="gyudon.jam" type="application/x-jam">
</OBJECT>
<A ijam="#gyudon" href="index.htm">ﾀﾞｳﾝﾛｰﾄﾞ</A>


</body>
</html>
```


★影さん，衰えませんね。（福岡市・**前田宣弘27歳**）……【**影**：その言葉はDr.にかけてあげたほうが，喜ぶよー。**Dr.D**：ワシを喜ばせてどうするの。**影**：うわーっ！最近，Dr.もお茶目になったって評判ですよ。**Dr.D**：だから，ワシを喜ばせてどう…。**影**：はいはい。】

iアプリ用



iアプリ版テキストエディタ

実用

iedit（アイエディット）

Mr. 親父（18歳）

内容

iアプリ対応i-mode用テキストエディタです。一番上に表示される20×5のテキストボックスで文章を編集します。



《写真1》iアプリでテキストエディタを再現



《写真2》当たり前だが、文字はしっかりと入力可能



《写真3》保存面での機能は充実

操作方法

「保存」,「開く」の方法として「携帯」,「Web」の二つの方法が使えます。「携帯」は携帯電話本体に保存・開くを行い,「Web」は専用のCGIを使って保存・開くを行います。

・携帯に保存する。

「File No」から1-5を選択し,「携帯に」を選択後,「保存」ボタンを押す。

注意：ひとつの「File No」に保存する文章の目安は,半角英数字で1000文字です。

・Webに保存する。

一番下に表示される20×1にテキストボックスに専用のCGIのURLを入力後,「File No」から1-5を選択し,「Webに」を選択後,「保存」ボタンを押す。

注意：保存する際には，文字数に応じたパケット料金が課金されます。

・携帯から開く。

「File No」から1-5を選択し,「携帯から」を選択後,「開く」ボタンを押す。

・Webから開く。

一番下に表示される20×1にテキストボックスに専用のCGIのURLを入力後,「File No」から1-5を選択し,「Webから」を選択後,「開く」ボタンを押す。

注意：開く際には，文字数に応じたパケット料金が課金されます。

プログラムの仕組み

iアプリ対応iモードには,「ScratchPad」と呼ばれる記憶領域があるので，ここを利用して，文章の「保存」を行います。

また，携帯電話は常にネットワークにつながってる状態でもあるので，Web（CGI）を利用して「保存」ができるようにしました。今回もi-jadeでデバッグしただけで実機では動かしてません。早く自分専用のi-modeが欲しいです。

いろいろ

今回，このプログラムを作るにあたって，中嶋康弘さんのホームページ「携帯電話でJavaしてみよう！「My iアプリ」」を非常に参考にさせていただきました。

しかし,「ScratchPad」・「ネットワク」を使ったのはいいんですが，iアプリでテキストエディタの需要はあるんでしょうか？

CHECKER FLAG

つぐ美：iアプリでの実用プログラム。説明文にもあるように，実用性に難はありますが，こういった形式のiアプリプログラムは少ないので，参考リストとして重宝するのではないでしょうか？

編：実機での動作には，スクラッチパッドに保存できないという不具合がありました。また，cgiの使用もこだわった作りではありますが，cgiの性質上，使用できるユーザーが限られてしまうかもしれませんね。

Dr.D：リスト面ではtextboxの使用など，いかにもJavaらしいリストと言える。この作品のメインとなる部分も，じつはこのtextboxの機能そのものだったりする。だが，つぐ美クンも言っておるように，こういったリストはiアプリプログラムを行ううえでは参考になるハズじゃ。

★パソコン＋免許＝約45万円！現在15万円…キツイ…。（大阪府河内長野市・**よっち18歳**）……【**編**：夏休みのアルバイトでもう少したまったでしょう。**Dr.D**：ワシの肩もんでも，時給2000円じゃしな。**影**：えっ？ではさっそく！モミモミ。**Dr.D**：いたたたた…。】

《リスト１》textedit.javaのプログラム・リスト

```
import com.nttdocomo.ui.*;
import com.nttdocomo.io.*;
import com.nttdocomo.net.*;
import java.io.*;
import java.util.*;
import java.lang.*;
import javax.microedition.io.*;

public class textedit extends IApplication {
        public void start() {
                Display.setCurrent(new SPPanel());
        }
}

class SPPanel extends Panel implements ComponentListener {
        Label label1,label2;
        TextBox textbox1,textbox2;
        Button button1,button2;
        ListBox listbox1,listbox2,listbox3;

        SPPanel(){
                label1 = new Label("File No.(1-5)");
                label2 = new Label("CGI URL");
                textbox1 = new TextBox("",20,5,TextBox.DISPLAY_ANY);
                textbox2 = new TextBox("",20,1,TextBox.DISPLAY_ANY);
                textbox2.setText(load("6"));

                listbox1 = new ListBox(ListBox.CHOICE);
                listbox1.setSize(24, 16);
                listbox1.append("1");
                listbox1.append("2");
                listbox1.append("3");
                listbox1.append("4");
                listbox1.append("5");

                listbox2 = new ListBox(ListBox.CHOICE);
                listbox2.setSize(65,16);
                listbox2.append("携帯に");
                listbox2.append("Webに");

                listbox3 = new ListBox(ListBox.CHOICE);
                listbox3.setSize(65,16);
                listbox3.append("携帯から");
                listbox3.append("Webから");

                button1 = new Button("保存");
                button2 = new Button("開く");
                this.add(textbox1);
                this.add(label1);
                this.add(listbox1);
                this.add(listbox2);
                this.add(button1);
                this.add(listbox3);
                this.add(button2);
                this.add(label2);
                this.add(textbox2);

                this.setComponentListener(this);
        }

        public void componentAction(Component c ,int type , int param){
                int list2_select = listbox2.getSelectedIndex();
                int list3_select = listbox3.getSelectedIndex();
                if(c == button1 && list2_select == 0){
                    save(textbox1.getText(),listbox1.getItem(listbox1.getSelectedIndex()));
                }else if(c == button1 && list2_select == 1){
                    save_connection();
                }else if(c == button2 && list3_select == 0){
                    textbox1.setText(load(listbox1.getItem(listbox1.getSelectedIndex())));
                }else if(c == button2 && list3_select == 1){
                        load_connection();
                }
                else if((c == textbox2) && (type == this.TEXT_CHANGED)){
                        save(textbox2.getText(), "6");
                }
        }

        public void save(String str,String File_No){
        try{
        DataOutputStream out = Connector.openDataOutputStream("scratchpad:///" + File_No);
        out.writeUTF(str);
        out.close();
        }catch(IOException e){}
        }

        public String load(String File_No){
        String str = new String();

        try{
        DataInputStream in = Connector.openDataInputStream("scratchpad:///" + File_No);
        str = in.readUTF();
        in.close();
        }catch(IOException e){}
        return str;
        }

        public void save_connection() {
                String str = new String("");
                String url = new String("http://127.0.0.1/mr_oyaji/cgi-bin/xxx.cgi");
                try {
                        HttpConnection httpconn =
(HttpConnection)Connector.open(url,Connector.READ_WRITE,true);
                        httpconn.setRequestMethod("POST");
                        httpconn.setRequestProperty("Content-Type",
"application/x-www-form-urlencoded");
                        OutputStream out = httpconn.openOutputStream();
                        byte[] data0;
                        data0 = (listbox1.getItem(listbox1.getSelectedIndex()) + "=" +
textbox1.getText()).getBytes();
                        out.write(data0);
                        out.close();
                        httpconn.connect();

                        InputStream in = httpconn.openInputStream();
                        ByteArrayOutputStream bytearrayout = new ByteArrayOutputStream();
                        byte data1[] = new byte[1];

                        int len;
                        while(true){
                                len = in.read(data1);
                                if(len < 0) break;
                                bytearrayout.write(data1);
                        }
                        str = bytearrayout.toString();
                        bytearrayout.close();
                        in.close();
                        httpconn.close();
                }catch(IOException e){}
                Dialog dg = new Dialog(Dialog.BUTTON_OK,"結果");
                dg.setText(str);
                dg.show();
        }

        public void load_connection() {
                String str = new String("");
                String url = new String("http://127.0.0.1/mr_oyaji/cgi-bin/xxx.cgi");
                try {
                        HttpConnection httpconn =
(HttpConnection)Connector.open(url,Connector.READ_WRITE,true);
                        httpconn.setRequestMethod("POST");
                        httpconn.setRequestProperty("Content-Type",
"application/x-www-form-urlencoded");
                        OutputStream out = httpconn.openOutputStream();
                        byte[] data0;
                        data0 = ("load=" +
listbox1.getItem(listbox1.getSelectedIndex())).getBytes();
```

★今年のプロ野球は，ヤクルトできまりですね，編さん。（群馬県前橋市・**はまちゃん14歳**）……【**編**：巨人もまた盛り返してきたしね。ま，どっちでもいいや。来年に向けた戦いに期待の中日です。**影**：ファンっていうのはうれしいもんだね。ベーマガファンもしかり。】

```
                    out.write(data0);
                    out.close();
                    httpconn.connect();

                    InputStream in = httpconn.openInputStream();
                    ByteArrayOutputStream bytearrayout = new ByteArrayOutputStream();
                    byte data1[] = new byte[1];

                    int len;
                    while(true){
                            len = in.read(data1);
                            if(len < 0) break;
                            bytearrayout.write(data1);
                    }
                    str = bytearrayout.toString();
                    bytearrayout.close();
                    in.close();
                    httpconn.close();
            }catch(IOException e){}
            Dialog dg = new Dialog(Dialog.BUTTON_OK,"結果");

            if (str != "LOAD ERRER" && str != ""){
                    dg.setText("LOAD OK");
                    textbox1.setText(str);
            }
            else{
                    dg.setText(str);
            }
            dg.show();
        }
}
```

《リスト２》textedit.jamのリスト

```
AppName = textedit
PackageURL = textedit.jar
AppSize = 2612
AppClass = textedit
LastModified = Sat, 02 Jun 2001 09:46:41
SPsize = 5200
UseNetwork = http
```

《リスト３》HTMLファイルの記述例

```
<html>
<HEAD>
<META http-equiv="Content-Type" content="text/html; charset=Shift_JIS">
<TITLE>textedit</TITLE>
</HEAD>
<body>


<OBJECT declare id="textedit" data="textedit.jam" type="application/x-jam">
</OBJECT>
<A ijam="#textedit" href="textedit.htm">ﾀﾞｳﾝﾛｰﾄﾞ</A>


</body>
</html>
```

《リスト４》cgiファイルのリスト

```
#!/usr/local/bin/perl
#
# textedit.cgi
#
# (C) Mr.OYAJI
#

if ($ENV{'REQUEST_METHOD'} eq "POST") {
        read(STDIN, $query, $ENV{'CONTENT_LENGTH'});
}
else {
        $query = $ENV{'QUERY_STRING'};
}

open (FILE, "textedit.dat");
        @data = <FILE>;
close (FILE);

@query_splited = split /=/, $query;

$user_agent=$ENV{'HTTP_USER_AGENT'};
$use_docomo = index $user_agent, "DoCoMo";
$use_java = index $user_agent, "Java";

print "Content-Type: text/html¥n¥n";

if ($query ne "" && @query_splited[0] ne "load"){
        push @data,"$query¥n";
        open (FILE, ">textedit.dat");
                $save_check = print FILE @data;
        close (FILE);

        if ($save_check == 1){
                $save = "OK";
        }

        if ($save eq "OK" && ($use_docomo >= 0 || $use_java >= 0)){
                print "SAVE OK";
        }
        elsif ($save ne "OK" && ($use_docomo >= 0 || $use_java >= 0)){
                print "SAVE ERRER";
        }
}
elsif ($query ne "" && @query_splited[0] eq "load"){
        @data_reversed = reverse @data;
        foreach (@data_reversed){
                $no_index = index $_, @query_splited[1];
                if ($no_index == 0){
                        $str_ = $_;
                        @str_splited = split /=/, $str_;
                        $str = @str_splited[1];
                        $load = "OK";
                        last;
                }
        }

        if ($load eq "OK" && ($use_docomo >= 0 || $use_java >= 0)){
                chomp $str;
                print "$str";
        }
        elsif ($load ne "OK" && ($use_docomo >= 0 || $use_java >= 0)){
                print "LOAD ERRER";
        }
}

if ($use_docomo < 0 && $use_java < 0){
        print "¥n";
        print "<html>¥n";
        print "<body>¥n";
        print "<p>";

        foreach (@data){
                print "$_<br>¥n";
                print "<hr>¥n";
        }

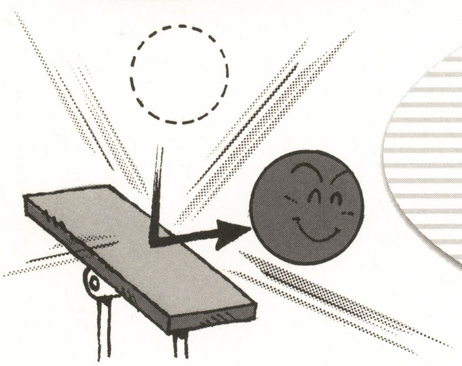
        print "<p>";
        print "</body>¥n";
        print "</html>¥n";
}

exit;
```


★「C言語講座」復活希望！「Dr.Dのベスト・プログラム研究室」拡大希望！おもしろかった記事を五つにしてください。（福井県武生市・つかち〜。16歳）……【Dr.D：読者からのいろんな意見，要望が今後のベーマガをますますおもしろくする！よろしく！】

(超大作プログラム)　リストが誌面に掲載できない作品です。
BMホームページ(http://www.basicmagazine.net/)からダウンロードしてお楽しみください。

ワンダースワンカラー(WonderWitch)用

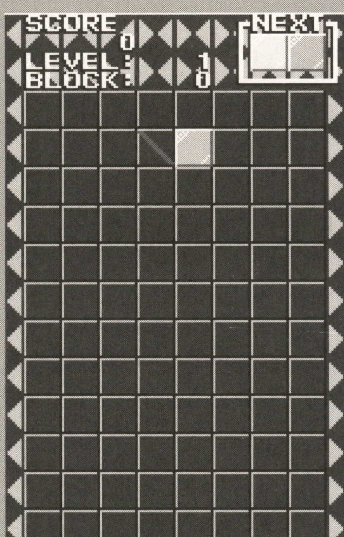


スワンカラーに投稿作品を移植

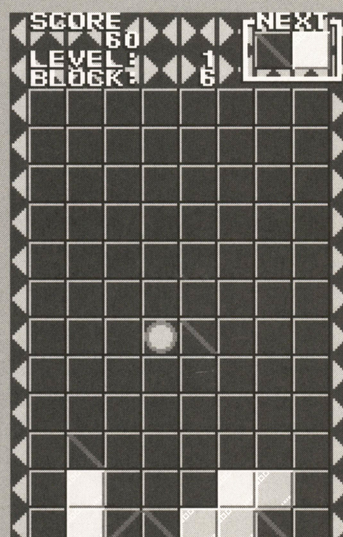
ミラーズ　フラッシュ

MIRRORS-FRASH

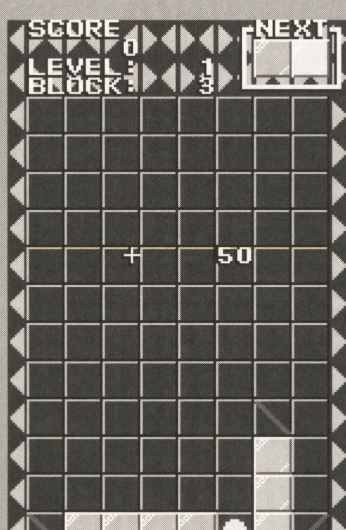
畑の木（28歳）



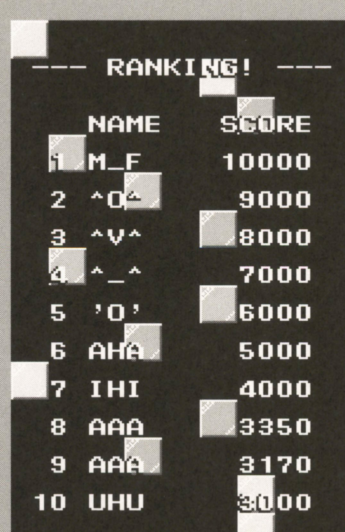
《写真１》実機では画面がカラー表示だ



《写真２》ミラーの扱いかたが鍵となるゾ



《《写真３》連続で消せば得点もアップ



《写真４》ランキング機能も搭載されている

はじめに

ベーマガ1994年10月号，MatsuriSoftさんの『MIRRORS-FRASH』をワンダーウィッチに移植してみました。C言語とBASICの言語の違い，機種間の仕様の違いなどから「グラフィックを簡略化してプログラムを短くまとめる」か，「プログラムはいくら長くなっても完全な移植を目指す」か，どちらにしようか迷いましたが，結局は後者を選択しました。そのため，移植にもかかわらず，とてつもなく長いプログラムになってしまいました。

で，大変申し訳ないのですが…大作扱いで…（汗）。それから，ワンダースワンカラー専用ということで…（汗）

遊びかた

原作となんらルールを変えていませんが，ちょっと古いゲームなので，きちんと説明を。

このゲームは落ちものパズルで，緑，黄色，水色の３色のブロック，同様に３色のフラッシュボール，そして２種類のミラーのいずれかが，２個１組で降ってきます。フラッシュボールはブロックをすりぬけて直進し，ミラーに当たると，90度向きが変わります。

このとき，通り過ぎたブロックとボールの色が同じなら，ブロックは消滅し，得点になります。また，ミラーに当たったとき，そのミラーも消えます。ボールは画面外に出ると消滅します。

★最近，昔のMSXを持ち出して遊んでいます。電源入れるとBASICが立ち上がります。感動！（親戚のお兄さんからのもらいもの）（青森県八戸市・C.C.C14歳）……【Dr.D：昔のものだろうがなんだろうが，感動することはいいことじゃ。影くんの場合は「勘当」。】

モードと操作方法

このゲームは，ワンダースワンを縦に持ってやります。Yボタンの左右でブロックの移動，下で高速落下です。Xボタンのいずれかで，パーツの左右の入れ換えができます。

モードは原作同様に三つ用意されていて，Yボタンの上下で選択，Xボタンでスタートです。

・**ノーマルモード**：ごくごく普通のモードです。

・**エビルモード**：邪悪なモードです。ブロックを一組置くたびに，なにかが降ってきます。

・**アナザーモード**：落ちてくるオブジェクトが，引っかかります。ただしフラッシュボールが出ると，引っかかっていたブロックなどは,すべて下に落ちます。

タイトル画面でSTARTボタンを押すとゲーム終了で，ゲーム中に押すと，タイトル画面に戻ります。

プログラムの起動方法

とりあえず，三つのfxファイル（ゲーム本体の"mirror.fx"，BGM用の"mf_bgm.fx"，効果音用の"mf_se.fx"）とsound.ilをワンダースワンカラーに転送してください。ランチャで起動できます。

また，ランキング用スコアファイルをram0領域に作成するので，2048バイト以上の空きを作っておいてください。

言いわけ・その他

なんで，プログラムがこんなに長くなったのか？単純に，私がプログラム下手なのが原因です。Cに精通した人なら，同じ内容でかなり短くできたと思われます（スイマセン，Cは6年くらい昔に，ちょっとかじった程度なので…)。

それから，キャラクターがめちゃくちゃたくさんあって，とくに縦画面用のフォントを作るのに苦労しました。

ついでに，どうせ長くなったんだから，BGMをつけようということで，イタリア民謡の『チリトンバ』なる曲をつけました。ゲームに合わないと思う方は，音量をさげるなり，鳴らさなくするなり，別の曲に変えるなりしてください。

最後に，原作者のMatsurinSoftさんに，敬意と感謝を…。

★参考にしたもの

・ベーマガ1994年10月号『MIRRORS-FRASH』(MatsurinSoft氏)

・ワンダースワンゲームプログラミング(長久勝著：ソフトバンク)

・プログラミング言語C（B.W.カーニハン／D.M.リッチー著：共立出版)

・今日から吹けるやさしいクロマチック・ハーモニカ（崎元譲　編著：龍吟社)

CHECKER FLAG

編：ベーマガとしては，初のスワンカラー作品です。ゲーム内容も過去の投稿プログラムの移植ではありますが，遊びやすい作品に仕上がっています。今年Witch投稿に期待している影さん，この作品はいかがですか？

影：実際のゲーム機で動作すると，やはり見栄えがするね。過去の投稿作品がワンダースワン上で動いているってのも，ある意味で時代を感じるものがある。

Dr.D：見栄えのする画面ではあるが，じつはこの作品はWonderWitchの特徴であるスプライトは使用していない。というのも，こういったパズルゲームの場合は，パネルの番号や配置などを記憶した配列を利用してscreenとして表示したほうがプログラムの効率がよいのじゃ。この手段はWitchのパズルゲームでは定番のような感じじゃな。

新連載！ 読者の疑問にプロのプログラマー，開発者が答える！ 「誌上公開質問状」連載開始！ みなさんからの，質問のお便りを募集します！

この新連載は，プログラミングに関して困っている点や疑問点，または製品購入に際しての疑問点，設定で悩んでいる点などを，メーカーのプログラマーや開発者のみなさんに回答してもらおうという企画です。この連載にあたり，みなさんからの質問・お便りを募集します。

①回答してもらいたいメーカー名

②疑問点（くわしく具体的に）をハガキまたは封書で下記までお送りください。質問の回答はすべて誌面で行います。

宛先：

〒141-8715 東京都品川区東五反田1-11-15　電波新聞社

マイコンBASICマガジン編集部「誌上公開質問状」係

メールの宛先：webmas@basicmagazine.co.jp

(件名を「誌上公開質問状」としてください)

※住所／氏名／年齢／職業（学生の場合学年）／連絡先電話番号／メールアドレスを必ず書いてください。

中学生・高校生を対象にした「ホームページコンテスト」応募締め切り迫る！

財団法人　国際コミュニケーション基金（ICF）主催の『第4回ICFホームページコンテスト』開催概要は，ベーマガ8月号，9月号で紹介していますが，作品の応募締め切りがせまってきました。個人のホームページ，クラブやクラスのみなさんで作ったホームページ，なんでもOKです。入賞者にはコンピュータ周辺機器も贈呈されます。どしどし応募してみましょう。

締め切りは2001年9月30日（日）だ！

●応募方法：

ホームページ上でURLを登録

http://www.icf.or.jp/contest/

●問い合わせ先：

e-mail contest@icf.or.jp　fax 03-5202-8509

★受験生になってもベーマガは読みつづけます。でも，受験生ってこんなに時間がないとは…つらい。(兵庫県神戸市・JANDY18歳)
……【編：そこを乗り越えて人間たくましくなってくんだから。影さんだって，同じ経験をして，今があるんだよ。影：その引用，複雑。】

FX-890P用



スピード感のある障害よけ

雨降り（あめふ）Pakupa（パクパ）

川口　周平（20歳）

《第1表》変数表

X	Pakupaの X座標
P	雨漏りの X座標
Q	雨漏りの Y座標
F	雨漏りの Y座標を調べる
N	棒グラフのキャラクター表示位置
S	雨漏りの落ちる速さ調節用
C	キャラクター作成用
TIMER	1/10秒ごとに1ずつ加算されるシステム変数
S$()	ゲーム終了時の文字列
H$()	棒グラフのキャラクターデータ
C$	キャラクター読み込み用
J,K	繰り返し用
V	押されたキーのコード
VM	V=31の値を代入したもの
L	雨漏りの落ちる速さ
D	雨漏りを処理しなければならない回数
Z	文字表示のために使用
A	雨漏りを処理できたか
I$	キー入力用
M	棒グラフのキャラクター指定用

ストーリー

Pakupaの家では雨漏りがひどい。屋根の修繕でもと思っていた矢先に，雨が降り出した。困ったPakupaは，雨漏りを食べてしまおうと考えた。

遊びかた・操作方法

RUNさせると，メニューが表示されます。上・下キーでレベルを，左・右キーで雨漏りの落ちる速さを選択し，スペース・キーを押すとゲーム開始です。

何かキーを押すとPakupaは左に動き，離すと右に動きます。左端の棒グラフは，雨漏りを処理しなければならない回数を表しています。うまく食べることができれば1ずつ減り，失敗すれば1ずつ増えます。すべて処理するとクリア，棒グラフが画面の上端をこえてしまうとゲーム・オーバーです。

プログラムの仕組み

IF文では使い勝手が悪いので，ON GOSUB／GOSUB文や比較演算子を用いた式を使いました。

具体的な入力方法

すべてBASICですので，普通に打ち込んでください。

苦労した点

画面を広く使おうといろいろ考えました。当たり判定のとき，BEEP音をつけられなかったのが残念です（BEEPの実行には，ずいぶん時間がかかるので，処理がもたついてしまうため）。

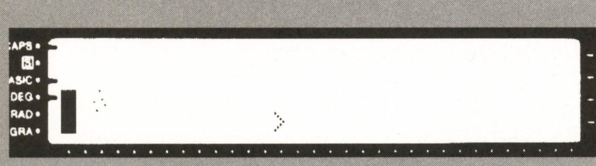
CHECKER FLAG

つぐ美：チョット久しぶりとなったポケコンプログラムです。iアプリ投稿などに負けないよう，ポケコンユーザーには頑張ってほしいものですね。

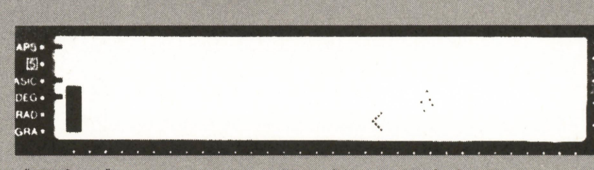
Dr.D：ゲーム内容は簡単なものじゃが，一つのキーで遊べる，ポケコンの横長の画面をうまく利用している，などチョットした工夫が盛り込まれておる作品じゃ。説明で述べているように，IF文がたった一つしか使われていないのが，リスト面での特徴じゃな。多少見づらいリストにはなっておるが，その分実行速度の向上につながっておるじゃろう。

『雨降りPakupa』のプログラム・リスト

```
100 CLEAR:CLS:X=16:P=1:F=-1:N=2:S=1:C=2:TIMER=0:DIM S$(1),H$(8):FOR K=0 TO 1:READ
C$,S$(K):DEFCHR$(252+K)=C$:NEXT
110 FOR J=0 TO 8:C$=RIGHT$(HEX$(C^J-1),2):FOR K=0 TO 5:H$(J)=H$(J)+C$:NEXT:NEXT
120 LOCATE 10,0:PRINT "Level [ ] ";CHR$(227);CHR$(229):LOCATE 10,1:PRINT "Speed [ ]
";CHR$(228);CHR$(230):LOCATE 8,3:PRINT "Push Space key!!";:LOCATE 0,3:PRINT CHR$(135);
130 *MENU:V=ASC(INKEY$):VM=(V=31):N=N-(V=30)*(N>0)+VM*(N<3):LOCATE 0,N+VM:PRINT
CHR$(135+103*VM);:LOCATE 17,0:PRINT RIGHT$(STR$(N+1),1)
140 S=S-(V=29)*(S>0)+(V=28)*(S<2):LOCATE 17,1:PRINT CHR$(77+(S=2)+(S=0)*5):ON -(V<>32) GOTO
*MENU:FOR K=0 TO 3:LOCATE 8,K:PRINT TAB(24);:NEXT:L=3+S:D=8*N
150 *LOOP:Z=NOT(I$<>"")*2:LOCATE X+Z,3:PRINT CHR$(32-220*(P=X+Z AND A));:LOCATE X,3:PRINT
CHR$(61+Z);:IF P=X AND A THEN A=0:GOSUB *ATARI
160 I$=INKEY$:X=X-(I$<>"")*(X>1)+(I$="")*(X<30):ON -(TIMER>=L) GOSUB *FALL:GOTO *LOOP
170 *FALL:ON -A GOSUB *ATARI:LOCATE P,Q-(Q=0)*4-1:PRINT " ";:P=(-P*NOT
F)+INT(RAN#*(-F*30))-F:LOCATE P,Q:PRINT CHR$(253-Q MOD
2);:F=(Q=3):A=F:Q=-Q*(Q<3)-(Q<3):TIMER=0:RETURN
180 *ATARI:LOCATE P,3:PRINT CHR$(237-A*3);:D=D+A:M=7-A-D MOD 8:D=D-NOT A:ON -(D<0) GOTO
*FINAL:DEFCHR$(254)=H$(M):N=N-(M=1)*A:LOCATE 0,N:PRINT CHR$(254);:ON -(D>31) GOTO
*FINAL:N=N+(M=0)*NOT A:RETURN
190 *FINAL:BEEP -(D>31):LOCATE 10,1:PRINT S$(-(D>31)):LOCATE 9,2:PRINT "Push Any
key!!";:I$=INPUT$(1):CLS:END
200 DATA 044001800800,-GAME OVER!-,080180100400,"  -CLEAR!-  "
```



《写真1》降りてくる雨粒を食べるゲーム



《写真2》ドットイート風のゲームでもある



CMM 試聴室

毎月みなさんの曲を紹介するコーナーです。募集要項をご覧のうえ，どしどしご応募ください。ここで紹介しているデータは，CMMの WebSite（http://www.basicmagazine.net/cmm/）に収録されています。曲への意見，アドバイスなども大歓迎します。

DRIVE ……自作

投稿者：とっしーさん（24歳）

E-Mail：toshio@binary-id.net
制作音源　MU80

ハードボイルドな曲をイメージして作りました

ハードボイルドな曲をイメージして作りました。サックスがリードを奏でる，少し堅い感じのフュージョン曲です。飽きが来ないように，短めにアレンジしてみました。

サビの部分では，イントロの半音進行の伴奏にのせて，ブルー・ノートを少し引っかけてみました。そのため，スケール選択に少し戸惑いましたが，最後には上手くまとまったのではないかと思います。

曲の後半にサックスとギターのソロがあります。あまり調性は気にせず自由に打ち込むことでアドリブ感を出すことを心がけました。また，ギターにエフェクトを使ったので，トラックをコピーし，少しずらせて貼り付けることでディレイ効果を補いました。

個人的には，イントロ後一番最初のフィル・イン（6小節目）が気に入っています。シンプルではありますが，左にふってあるフロア・タムとChineseシンバルの組み合わせが，いい感じにキマったと思います。

編集部から

緊張感のある進行で，ベースの動きが小気味いい曲です。楽器の構成やフレージングもうまくまとまってます。ベースとともにドラムの雰囲気作りもいい感じですね。全体的に，まとまりがあるという印象の強い曲です。

曲の構成でみていくと，A，B，Cとパートが進むにつれてどんどんまとまりがなくなっていきます。とくにAパートでかっこいいフレーズだったサックスがこころもとないです。Aパートでまず聴かせて，Bパートは中休み，Cパートはサビですので，Aに負けないくらい印象的なフレーズを入れてあげると曲がひきたつと思います。また間奏のサックスとギターのバトルですが，両方とも同じような音域をうろちょろしていて印象が薄いです。ギターの最後の部分が音が上がっていっていい感じなのですが，音量的に目立たないのが残念ですね。上がり下がりはある程度なめらかに，それでいてバトル部分が互いがどれだけ目立つか競う部分ですので，ハデに演出してみてください。

打ち込みでは，リバーブがちょっと強いかなと思いましたが，それ以外はいい音が出せていると思います。

ビー玉の瞳 ……自作

投稿者：そめっとさん（18歳）

URL：http://www2.ocn.ne.jp/~somet/
制作音源　VSC-88

詩をもとに作りました

同名の詩をもとに制作。「コドモからオトナへの通過点で感じるコドモ世界への執着」とかそんなような内容だった気がします。曲自体はいたってオーソドックスな内容です。唯一，ときおり，小節の最後で変拍子があらわれるくらいです。メロ自体もありふれた感じです。なにかノスタルジックな感じを出したかったです。

編集部から

ノスタルジックな感じ，という部分が，楽器の選択やフレーズにうまく出せていると思います。よくいう民族音楽系の響きとでもいうのでしょうか。聴きやすくまとまっている作品ですね。ここまでまとまっていると，Aメロのピッチカート奏法のストリングスとギターの定位が重なってたり，ドラムだけ妙に浮いて聴こえる，あと終わりかたが中途半端なのが残念です。

まとまっているというのは，逆の言いかたをすれば，個性がない，という意味にもなります。「いたってオーソドックスな内容」「メロ自体もありふれた感じ」と自分で感じるならば，それをなんとか変えてみたいという発想がほしいです。結局のところ，その，もとにしたという詩も，いたってオーソドックスな内容で，ありふれた感じだったのでしょうか。

打ち込み的な部分では，楽器の位置をはっきりさせてごちゃっとした感じをなくして，ドラムをもう少し抜けのいい音色にすると，すっきりした感じになるのと同時に，それが曲の特徴の一つとして生きてくると思います。曲自体では，変拍子が埋もれていてアクセントになっていないので，もう少し強調してみてもいいと思います。

マイコンBASICマガジン

明日のスター☆プログラマー

今月のベスト・プログラマー

影：今月は全体的にアイデア面で，一歩抜きん出た作品が見あたらないなぁ。今回は悪い意味で，ベストの選考に悩みそうですね，Dr.。

Dr.D：確かにそうじゃな。ただ，作品の完成度で言えば，超大作ではあるが，Visual C++用の『P.P.Year 7』と『必殺ぷれいや～』，あとWonderWitch用の『MIRRORS-FRASH』はなかなかのものじゃろう。

つぐ美：ですが，やはり影さんの言うとおりアイデア面でのマイナスが目に付くんですよね。『必殺ぷれいや～』形式のソフトは，インターネット上などで数多くあるし，『MIRRORS-FRASH』は昔の作品ではありますが，移植版なのでやはり新鮮味に欠けますね。

影：『P.P.Year 7』もいい作品なんだけど，1Pモードがないのが痛いよな～。やはり2人専用というのは，完成度の面でも中途半端な印象を与えちゃうね。

編：プログラム面でおもしろい作品となると，VB+DirectX用の『迷路』でしょうか。3D形式の表示やサウンドの機能を取り入れた投稿作品というのは，なかなかないので，そのぶん印象にも残りました。

つぐ美：今まで挙げた作品はすべて超大作プログラムね。一般投稿の作品にも目を向けてみると，やはり目に付くのはiアプリの投稿ね。実用プログラムの『iedit』もいいけど，わたしは『牛丼道』のほうがお気に入り。こういう単純なゲームのほうが，iモード上で遊びやすいんですよ。

影：牛食えのキャラクターを使用しているので，いかにもベーマガらしい作品と言えるし。

Dr.D：確かに『牛丼道』はなかなかの完成度じゃ。つぐ美くんの言うとおりゲーム内容は単純じゃが，そこをしっかりiアプリ上でゲーム化するというのも，ある意味ではアイデア賞ものじゃな。iアプリということもあるが，リストも比較的コンパクトだし，今月はこの作品がベストにふさわしいじゃろう。

編：というわけで，今月のベスト・プログラマーは，iアプリ用『牛丼食えスト外伝　牛丼道』の作者，ヅラChuさんに決定しました。

影：さてさて，ベストに選ばれた作品は，読者からの反応も好評な「Dr.Dのベスト・プログラム研究室」でくわしく解説しているゾ。そっちも忘れずにチェックしてくれ！



◀iアプリ用『牛丼食えスト外伝　牛丼道』

投稿ありがとう

MONOMON／中島光治／小泉賢一／BlueCube／西野紀之／橘　一馬／飯村世紀／池添貴之／ヅラChu／らーまそふと／大津善郁／Mr.親父／畑の木／金子拓哉／ていこく／宮西和機／渡辺栄彦／峠　恒司／清家　巧／Gs／金井義男／大熊通彦／福ちゃん／木下太郎／高橋和那／小坂文寛／金子宣之／赤星公昭／広瀬晴吉／たくちゃん／澤山慎一／ARAKI／西畑規剛／金泉貢／スタッツ／坂元　肇

※全員紹介できなくてごめんなさい

影さんのマジック・ボックス

ワンテク&アイデア募集コーナー

このコーナーは，みなさんのプログラミングに関するワンポイント・テクニックや，アイデアなどを紹介していこうというもの。投稿する際は，機種や動作環境などもしっかり明記してください。

あて先：〒141-8715　東京都品川区東五反田1-11-15　電波新聞社マイコンBASICマガジン編集部
「影さんのマジック・ボックス」係
e-mail：webmas@basicmagazine.net
件名に，影さんのマジック・ボックス係と記入してください。

iアプリの機種依存を改善する

iアプリ対応の503iは，画面サイズや文字サイズなどが機種ごとにバラバラ。たとえばSOは利用可能な画面サイズが120×120しかないし，Dの文字サイズは他より一回り大きいし。FやPでしか正しく見えないようなiアプリを見ると，Dユーザーとしては非常に嫌な気分になります（悪いのは互換性を持たせていないメーカーですが）。

そこで，おいらなりに見つけた，画面表示を（今のところ）全機種共通にする方法を示します。CanvasのPaint(Graphic g)メソッドに以下のソースを挿入してください。

```
//画面サイズの機種依存をなくす。以降は描画領域を
//(0,0)-(120,120)にすること。液晶画面の中央に表示されます。
g.setOrigin((getWidth()-120)/2,(getHeight()-120)/2);

//フォントサイズの機種依存をなくす。f.getAscent（フォント縦幅）が
//実機では12，i-JADEでは13に統一される（はず）
Font f=Font.getDefaultFont();
if(f.getAscent()>13){
 f=Font.getFont(f.SIZE_SMALL);}
g.setFont(f);
```

これを使って，どの機種でも正しく見えるiアプリを作っていただけたら幸いです（そういえば，まだDとSOはエミュレータ出てないんですよね…）。

by ヅラchu

投稿プログラム・コーナー
Programmer's Room

投稿ありがとう

編：超大作では，凝った作品が目立ちましたが，一般投稿の作品は採用作品も含めて，アイデア・内容ともにさみしい感じでした。最近の投稿で多いのは，「もう一歩踏み込めばよりおもしろくなるのに」という作品ですね。たとえば，ゲーム作品でいうと，操作などシステム部分はしっかり作られているのに，ゲームとしての内容に乏しいというもの。プログラム技術面だけでなく，作品の内容面でのこだわりも，これからの投稿作品にはほしいところですね。

投稿に関する読者からの質問

今月はプログラム投稿に関する読者からの質問の解答をまとめてみました。投稿の参考にしてください。

Q.ベクターなど，ウェブ上で掲載されている作品の投稿はいけないのですか？

A.投稿者自身のホームページ上で発表している作品の投稿はOKです。ベクターで発表している作品の投稿も，基本的にはOKですが，ソース・ファイルも一緒に収録したものを発表している場合，その作品の投稿はご遠慮ください。

ただ，できればこれらに掲載されている作品は，ベーマガ投稿用に改良を加えたものが望ましいです。

また，ほかの雑誌に投稿した作品の投稿は二重投稿となってしまうため，絶対におやめください。

Q.投稿した作品にバグがあったので，再投稿をしたいのですが？

A.修正版の投稿も受け付けていますが，その際には，修正した個所を記述のうえ，修正済み作品を新たに投稿してください（つまり，原稿，プログラムを添えて再投稿してください。修正個所のみの投稿はお止めください）。

また，バージョンアップに関してですが，バージョンアップごとにその作品を投稿される人もいますが，大幅な変更でない限り，バージョンアップ版の投稿はご遠慮ください。大きな変更を加えた作品は，新規の作品として投稿してください。

Q.オリジナルに作成されたプログラム言語の投稿はOKですか？

A.言語の制限はありませんが，オリジナル言語の場合，その言語の使用規定をよく読んで，再配布などの条件を確認してから投稿してください。また，投稿の際には，その言語の入手先を記述しておいてください。

一口CHECKER FLAG

●Visual Basic用『暗号化 Ver T-1.00』／Takuya（13歳）

●Visual Basic用『バイナリ暗号』／渡辺栄彦（16歳）

Dr.D：共に文章を暗号化するプログラム。最近投稿数が多いジャンルなだけに，アイデアやリスト面で抜きん出たものがないと採用はキビシイ。過去に掲載された作品と同じような作品では，新鮮味にも欠けてしまうゾ。

●Visual Basic用『いまはいつ？』／とっしー（12歳）

つぐ美：内容が基本的すぎるので，採用はむずかしいわ。VBの機能はまだまだいろんなことができるので，もっとプログラムの勉強を重ねて，こだわった作品を作ってね。がんばって！

●Delphi用『まるゑ』／らーまそふと（22歳）

編：コンパイラのパスを設定できる機能を持ったテキスト・エディタ。使う前のパス設定が，もっと簡単にできるとより手軽に使用できると思いますよ。

●HSP用『Rename Card』／大津善郁（17歳）

編：文字を削除するファイル名の指定場所を，前からだけでなく，好きな場所を指定できると使い勝手が上がるハズです。

●Visual Basic用『シューティング。』／清家　巧（16歳）

つぐ美：砲台からミサイルを発射するタイプのシューティング・ゲーム。操作はスクロールバーでなく，キーボードの方向キーのほうが遊びやすいハズだわ。あと，説明ファイルの名前や住所の記述を忘れないでくださいね。

●Visual Basic用『草刈り』／清家　巧（16歳）

影：基本はもぐらたたきだけど，敵を雑草に置き換えたもの。見栄えのする形式だけど，ゲームとしてはもう一つアイデア不足。あと，送られてきたソース・ファイルからの実行ができなかったゾ。

●Visual Basic用『LogoChanger』／中島光治（27歳）

Dr.D：Windows起動時のロゴを変更するソフト。定番のウラ技なので，おもしろ味に欠ける。また，せっかくソフト上でこの行為を行うのに，バックアップ機能がしっかりしていないのはいただけない。あと，メール投稿の場合はLHA圧縮を忘れずに行ってくれ。BMPなどがたくさんあったため，ダウンロードに時間がかかってしまうゾ。

●VBScript+DirectX用『サッカーイメージパネル』／金井義男（39歳）

つぐ美：以前にもう一歩だった作品の再投稿版。簡単なPKゲームが加わるという修正がありましたが，基本的には変わっていない印象ですね。

今月はもう一歩でした

●Visual Basic+DirectX用『タイピング』／西野紀之（13歳）

影：基本的なタイピング・ゲーム。遊べる作品に仕上がってはいるけど，出題される問題が少なかったり，演出面でさびしいものがあったりなど，ゲームとしてもう一歩の出来だったな。タイピング・ゲームについては，連載の「Catlikeのもっと身近にプログラミング」で取り上げているので（5月号，6月号，8月号で掲載），ぜひとも参考にしてください。

Dr.Dの ベスト・プログラム研究室

BEST PROGRAM LABORATORY

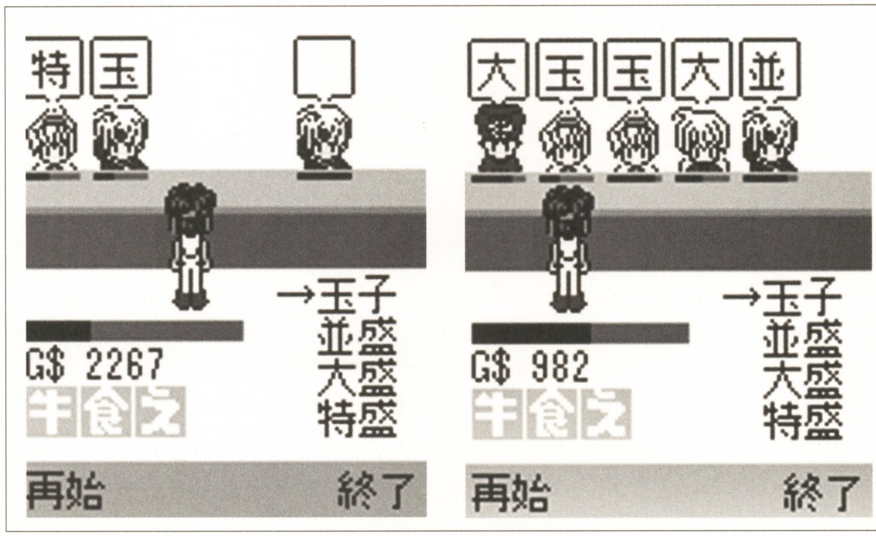


今月のベスト・プログラム：iアプリ用『牛丼食えスト外伝　牛丼道』(71ページ掲載)

読者からのアンケートで「ぜひ，これからも続けていってほしい」など，ありがたい意見をいただいておるこのコーナー。これからも読者の参考になるようなコーナーにしていきたいと思うので，引き続き応援を頼むゾ。

さて，今月のベスト・プログラムは，iアプリ用の『牛丼食えスト外伝　牛丼道』。投稿数も徐々に増えてきている注目のiアプリなので，読者も興味があるプログラムだと思う。ただ，iアプリはリストの構成も比較的単純なため，特別な処理をしている部分は少ない。というわけで，今回はゲーム開始前の設定部分(gifファイルの読み込みなど)と，ゲーム・システムの部分を解説していこう。

(Dr.D)



▲玉子メニュー追加改造画面

ゲーム開始前の設定

この設定部分では，iアプリならではの処理となるgifファイルの設定と，ゲーム画面を描画するpaintメソッドについて重点的に解説しよう。

まずは，gifファイル部分の設定。このプログラムでは，getImgというメソッドを自分で定義して，その中でgifファイルを使用する設定をすべて行っておる。gifファイルの使用には，MediaImageインターフェースを使用する。getImageメソッドを使ってgifファイルを読み込み，Image型のフィールド(変数)に取得すれば画像データとして使用できる。

だが，ここで注意したいのは，MediaImageに対して，画像ファイルを利用するという処理を行わなければならないという点。その処理がtry～catch文のmi.use()というわけじゃ。このgetImgメソッドは，gifファイル使用の処理をうまくまとめているので，ぜひとも他の作品へ流用したい処理じゃな。

ゲーム画面の描画は，paintメソッドで行う。g.lockといったpaintメソッド特有の命令を除けば，見た感じだけで，行っている処理がわかる命令ばかりじゃ。ゲーム画面で目につくカウンタやゲージも，fillRectといったGraphicsクラスで用意されているメソッドで描画しておる。

特徴的なのは，フォントサイズの調整を行っている点じゃな。これはムリに行わなくてもよいのじゃが，やはり多くの機種で遊べるようにするためには，必要な気遣いと言えるじゃろう。

《第1表》玉子メニュー追加の改造手順
String[ ] gsNameの最初に"玉",を追加
subInitメソッド内のgsStatusに入れる値を4にする
paintメソッド内の並盛表示の前に，g.drawString("玉子",88,81);を記述
"→"表示のY座標の指定をmyItem*12+81とする
//吹き出し部分のdrawStringの前にif(gsStatus[i]>=0)を記述
gsStatus[ ]<3となっている部分をすべてgsStatus[ ]<4とする
myItem=2，myItem>2となっている部分の数値を3にする
1,2,3キー入力時のmyItemに入れる値をそれぞれ1，2，3とする
(並盛の場合はmyItem=1となる)
//商品を出す，という部分を《リスト1》のように記述する
//お客の処理，という部分を《リスト2》のように記述する
新しいお客処理のgsStatus設定部分の%3を%4にする

ゲーム・システム

このゲームの流れを処理しているのはrunメソッド。このrunメソッドは，大まかに言うと，キー入力による主人公の移動，商品の選択，商品を出したときの判定，各お客の処理，新規のお客の出現処理，終了関連の処理を行っておる。

画面描画は行っていないの？と思った読者はなかなかスルドイ。画面描画はrunメソッドの最後にあるrepaint()によってpaintメソッドを呼び出すことで実行している。このため，このrunメソッドでは，描画に関する変数の処理を行えば，

それが自然と画面に反映されるわけじゃ。

キー入力の部分は，主人公の位置を示すmyXや商品を示すmyItemを，対応したキーによって変更しているだけなので，わかりやすい処理じゃろう。ただ，連続でキー判定されないような工夫を見逃してはならない。iUpやiRightといった変数の値(1or0になる)で，その処理を行っておる。

商品を出す部分も，出した商品とお客の希望商品（それぞれmyItemとgsStatusで示される）が，合致しているかを判断し，状況にあった変数の設定を行う。

お客の処理は，for文でカウンタの人数分(6人)の処理を一度に行う。ここでは主に，お客の待ち時間切れ(gsTimeが0)の時の処理を行っている。

新しいお客の処理は，乱数を使って出現させるか否かを判断している。出現させる条件になったら，待ち時間や注文するメニューの設定などを行う。

お客の基本データを設定したあとに，switch文でお客が誰なのかを判断し，特別な動きをするキャラクターごとに(ここでは，ケンタ，マイ姉，大王)設定を行う。お客に新たな個性を持たせたいときは，ここの部分を改造すればOKじゃ。

終了処理は，評判ゲージの値（myPop)によって判断しておる。また，キー入力による終了や再始判定も行っておるゾ。

改造してみよう

iアプリということで，画面構成を大きく変更する改造には向いていない。そこで今回の改造は，商品の追加というゲーム内容に関連する改造を行ってみた。追加するメニューは，牛丼には付き物の“玉子”。といっても玉子だけを注文するお客は不自然なので，玉子を注文したあとに牛丼を注文してくる，という動作を取り入れてみたゾ。

例によって改造したリストすべてを掲載するのは誌面の関係でムリなので，改造の手順と，文章説明ではわかりにくいと思われる部分の改造リストのみを掲載してみた。これを元にぜひとも改造にチャレンジしてみてほしい。また，iアプリ・プログラムの手始めとして，この改造にチャレンジするのもよいじゃろう。

≪リスト1≫商品を出す部分の改造リスト

```
//商品を出す
if((iKey==(1<<Display.KEY_SELECT))&&(gsStatus[myX]<4)&&
(gsStatus[myX]>=0)){
        if(myItem==gsStatus[myX]){//希望商品だった場合
                if(myItem==0){//玉子だったときの処理
                        myMoney+=50;
                        //牛丼を頼むまでの待ち時間設定
                        gsStatus[myX]=-20;
                        gsTime[myX]=1000;
                        gsCount[myX]=0;
                }
                else{//玉子以外で注文通りの商品時の処理
                        gsStatus[myX]=4;
                        myMoney+=10*myItem+29;
                        gsTime[myX]=20;
                        gsCount[myX]=1;
                }
                myPop++;
                if(myPop>100){
                        myPop=100;
                }
        }
        else{   //間違っていた場合
                gsStatus[myX]=5;
                myPop-=gsDown[myX];
                if(myPop<0){
                        myPop=0;
                }
                gsTime[myX]=20;
                gsCount[myX]=1;
        }
}
```

≪リスト2≫お客の処理部分の改造リスト

```
//お客の処理
for(i=0;i<6;i++){
       if(gs[i]!=0){
       gsTime[i]-=gsCount[i];
       if(gsStatus[i]<0){//玉子受け取り後の待ち時間処理
             gsStatus[i]++;
             if(gsStatus[i]==-1){
             //玉子のあとに注文する牛丼の設定
                     gsStatus[i]=Math.abs(rnd.nextInt()%3)+1;
                     gsTime[i]=1000;
                     gsCount[i]=myMoney/500+10;
             }
       }
       if(gsTime[i]<=0){ //待ち時間切れの判定
               if(gsStatus[i]<4){//商品待ちの場合怒って帰る
                       gsStatus[i]=5;
 (以下は同じため省略)
```

Catlikeの もっと身近にプログラミング！

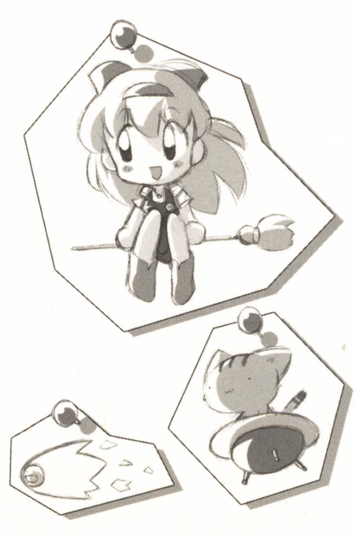
第5話 シューティング・ゲーム編 その2 オブジェクトってのを使ってみようじゃないか

自機と敵機のプログラムって，まったく別々に組んでしまいがちですけど，ちょっと待って。結構共通する場所ってあるんじゃない？　今回の方法は，プログラムを短く素早くわかりやすく組むにはぴったりです。

要は,心構えの問題だよね

月がきれいな季節になりました。ふと空を見上げると，そこには白い，真ん丸い，でも自分では決して輝くことのできない天体が。闇という神秘な時間と相まって，よく物語の引き合いに出される，おそらくは世界で一番純粋な演技者。彼を見上げて，多くの語り部がさまざまな物語をつむぎだしてきたことを感じます。そのたびに彼は，色仮面を身にまとい，時には悪役となり，舞台を照らし出してきたのでしょうか。

月にまつわる話はさまざまです。それこそ，数多くの逸話がありますし，それらを聞いてしまうと，月を見つめて新たな物語を思い浮かべてしまったりします。



月見は遠い昔から語り継がれてきた伝統的な儀式です。それゆえ，忙しいときには軽視しがちではありますが，ほんの少し知識を蓄えて迎えると，昨年とは別の月が見えるかもしれませんよ。

メインループをちょと復習

何の話かというと，月がある宇宙は，シューティング・ゲームの格好の舞台だなぁと。

さて，シューティング・ゲーム編は，ゲーム・プログラムの基礎とも言えるメインループについて考えていきます。このメインループを組んでしまえば，シューティング・ゲームであろうとなかろうと，プログラムをはじめるにあたっての地盤が完成するわけです。プログラムを組むときに一番面倒なのが，ハードウエアやらOSやらにプログラムを開始することを伝える手続きを組むこと。

Windowsソフトで例を出すなら，肝心のプログラム部分に入る前の手続きのために，何百行というコードを書かなくてはいけないのが当たり前となっています。ですが，メインループをあらかじめ作っておけば，この手続きなど面倒な部分は一気に飛ばしてしまって，いきなりゲームのプログラムに入ることができるわけです。うーん，なんて現実的かつ理想的。

そんな手の届く夢を追いかけつつ，前回はメインループの「ループ」たる所以をお話したところ。なんとなーく，メインループの全体像はつかめたのではないでしょうか。

で，前回は，おもいっきし何か重要な部分を抜かした話をしました。**第1図**を見てください。前回の最後に出てきたフローチャートです。

見れば一目瞭然。あっさり「計算処理」とか「描画処理」とか書いてブロックにしてますけど，これ，中は実際何をやればよいのかまったく書いていません。そりゃー，ゲームによって何をやるかなんて，さまざまなのですが，わざわざ「シューティング・ゲーム編」なんて看板ひっさげてるわけですから，少しは絞り込んだお話しないとなー，なんて思うわけですよ。

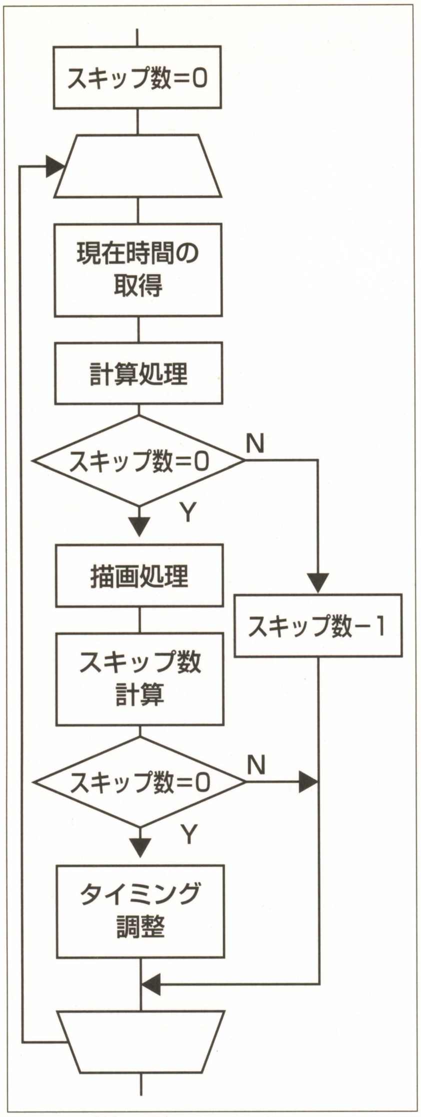
で，このどちらを優先するかといえば，見た目的には「描画処理」のほうが画面にすぐ結果が出るしわかりやすいし，組

筆者紹介

catlikeホームページ
http://catlike.marchen.to/
某所でこっそり「いつ何処。」のプレビュー版を公開しました。Web上で配る日はくるかな……？

文:まかべひろし
奮起して部屋の掃除を行いました。ボクの青春時代を支えてくれたMSX関連の資料やらなにやらをまとめて処分しました。総重量300kg以上。でも,それによって培ったもののほうがよっぽど重い気がするのです。

絵:桜部のとす
ツカレター。今回の電車の絵はかなり気合い入れて描きました。とりあえず今現在の全力。もとはカラーで描いたんですよ～。ココだと白黒になっちゃいますけどね（＾＾；



《第1図》フローチャート

んでいて楽しいのですけれど，ゲームとして考えれば優先すべきは「計算処理」のほう。何か計算をした結果が，画面に絵として反映されるわけですから，その結果を導き出す部分が欠けてしまっていれば，描画処理を組めるわけがありません。ここは一つ，お楽しみは最後に取っておくという子供のおやつのような考えかたで，うっとうしい計算処理のほうを乗り切ってしまうことにしましょう。

ゲームはパーツの集まりだ

シューティング・ゲームで計算処理というと，どんなものが思い浮かびますか？　まずは，既存のゲーム，もしくは自分が作りたくて思い描いているゲームを想像してください。まず，自機が中心にいますね。その周りには敵機がいて，個々のキャラクターはキー入力や，あらかじめ決められた軌道どおりに動いています。時には弾を発射しています。アイテムが落ちていて，それを取りにいかなければいけないかもしれません。

それらの動きや衝突判定，弾の発射やアイテムの効果など，ゲーム中に起こりうるすべての事象が「計算処理」に入ってくるのです。もちろん，個々の動きやアイテムの効果などはゲームによってさまざまであり，これを工夫し組み合わせ，今までに見たことのないゲームを作り出すことがゲームクリエイターと呼ばれる人たちのお仕事なのですが，ここではその「新しいこと」はとりあえず考えないでおくことにします。

そうすると，おおよそのシューティング・ゲームで同じような動作をしている部分が見えてきます。自機はキーの操作によって動かされ，敵機はプログラムの計算によって動かされています。弾などの動きも同様，あらかじめ決められた動き，たとえば直進だとか，自機を追いかけてくるような誘導的軌跡であるとか，敵機と同様に処理することができそうです。

さて，当たり前のようにここまで書いてきましたが，これらは一つの前提のもとに成り立つ仮定です。プログラムに慣れた人ならごくごく自然に考えてしまうことなのですが，1台のモニタに表示されるゲーム画面は，複数のパーツによって構成されています。この「パーツで構成されている」というのが，とてつもなく重要。画面に表示されるパーツ数やタイプはちがえど，一つの「パーツ処理ルーチン」及び「パーツ描画ルーチン」を通れば，画面が構成できることになります。

実際にプログラムを組むことを考えると，パーツを一つ処理するのも，パーツを10個処理するのも，似たような処理を行うとすればその回数分だけループを組めばいいわけですから，プログラムを組むというえでの負担量はさほど変わらないことになります。

このパーツというもの，ここではそれをカッコつけて，オブジェクトって名前で呼んでみることにします。プログラム用語のオブジェクトとはちょっとちがいますけど，ゲーム中のパーツという意味では正しいですからね。

描画用の情報をオブジェクトとして

シューティング・ゲームにおけるオブジェクトっていうのは，どんな情報で構成されているのでしょうか？　自機や敵機，そしてそれらから発射される弾がオブジェクトであるとして，ちょっと考えてみましょう。

共通されるのは，どれもこれも「目に見える情報である」ということですね。目に見えるものを表すために必要な情報，って考えると，比較的簡単に思いつきやすいですね。ビットマップ画像を画面に表示するためには，

- **元画像**
- **転送元の座標（左上と右下）**
- **転送先の座標（左上と右下）**
- **転送方法（ラスタオペレーション）**

という情報が必要になります。

元画像は，敵機の絵などが描かれたビットマップ画像です。これは表示前に，あらかじめ読み込んでおく必要があります。

転送元の座標は，1枚のビットマップ画像の中に複数のパーツが含まれている場合などに必要になります。パーツのアニメーション・パターンは1枚のビットマップ内に規則正しく配置することが多いので，転送元座標は必須ともいえます。転送先の座標は，最終的にモニタに表示するという前提のもとに必要になります。もし転送先座標がなければ，モニタのどこに表示をしてやれば良いのかまったくわからなくなってしまいます。移動という処理は，ここの値を連続的に書き換えていくことで行われます。

最後の転送方法というのは，各プログラム環境に依存してしまいますが，たとえばWindowsのGDI描画では元の画像に論理演算（ANDとかOR）を処理しながら転送することができます。また，



《リスト1》描画に必要な情報を含めたオブジェクトを10個作るコード

```
// 例は、Windows環境下、DirectDrawを用いようとした場合

// オブジェクトの定義
typedef struct {
    LPDIRECTDRAWSURFACE    pDDS;      // Bitmap
    RECT                   rcSrc;     // 転送元座標
    RECT                   rcDst;     // 転送先座標
    DWORD                  dwDrawFlg;// 転送方法
} OBJDATA;

// オブジェクト配列の実態
OBJDATA pObjData[10];

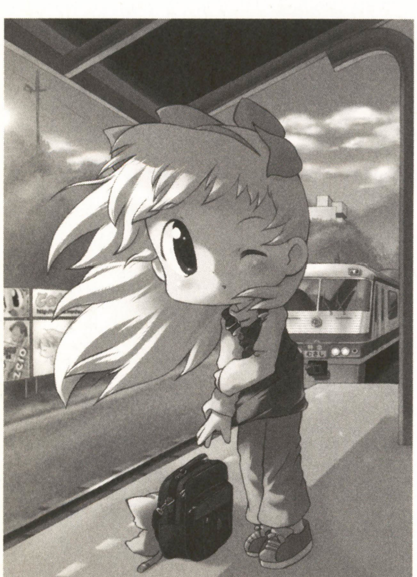

※WindowsやDirectDraw開発を知らない人のための解説。
・LPDIRECTDRAWSURFACE は、画像を入れておくための「サーフェイス」と呼ばれる場所へのポインタです。
・RECTは左上座標を(left,top)、右下座標を(right,bottom)として格納する構造体です。以下のように定義されています。
typedef struct {
    int    left,top,right,bottom;
} RECT;
・DWORDは unsigned long と同意です。
```

PlayStationなどの家庭用ゲーム機ではαブレンドと呼ばれる半透明処理を行うことができます。DirectXを搭載している環境であれば，透明色を指定して転送することも可能です。

これら複雑な転送方法も，情報としてもつ必要があるでしょう。

これだけの情報が「目に見えるオブジェクト」には必須となります。反対に，これだけの情報があれば，移動という計算処理と，描画の処理を行うことができるわけで，お粗末ながらシューティング・ゲームを構築することができるかもしれません。

さて，これらのオブジェクト情報をプログラム上ではどのように管理してあげればよいでしょうか。情報を管理するのは，決まってメモリの中となるでしょう。C言語であるなら，構造体と配列を用いれば簡単に表現することができます。シンプルなBASIC言語でも，配列をいくつか作ってやることで，この情報を管理することができます。



例を出してみましょう。前述の，描画のために必要な情報を含めたオブジェクトを10個作る場合，**リスト1**のようなコードを書けばよいでしょう。

この例では，OBJDATAという構造体を定義し，その中にオブジェクトとして必要な情報を定義しています。これでオブジェクト一つ分の情報はできたので，それをpObjData[10]という形で，10個分のオブジェクトの情報格納場所を作りました。ちなみにC言語の場合は，配列の個数は実行時に変更はできませんので，あらかじめ多めに作っておくか，malloc()などの動的メモリ確保を用いて対処する必要があります。BASICの場合も，あらかじめ配列を多めに確保するのが良いでしょう。

実際に値をセットするときは，

```
pObjData[0].dwDrawFlg = 0;
```

のように，配列の[]の中に操作したいオブジェクト番号を入れます。こうすることでオブジェクトの情報にアクセスできるので，つまりは，

```
for ( i = 0; i < 10; i++ ) {
  pObjData[i].dwDrawFlg = 0;
}
```

のように，ループでまとめて処理をすることもできるわけです。オブジェクトの個数を増やしたいときは，10という数値を任意に書き換えればいいだけなので，非常に柔軟なプログラムと言えますね。

メインループ内「計算処理」では，このようにして作ったpObjData[n]のそれぞれの項目に必要なデータを入れてやります。たとえば右に6ドット移動する，という処理ならば，転送先のXを＋6してやればいいので，

```
pObjData[i].rcDst.left += 6;
pObjData[i].rcDst.right += 6;
```

と書けばよいことになります。

メインループ内「描画処理」では，こうして操作したオブジェクト・データを参照しつつ，実際の描画を行っていくことになります。こちらもオブジェクトの数だけループを組んでやり，その中で一つ一つ転送するということになりますね。

ゲーム性もオブジェクトで表現！

なんとなーく形が見えてきたでしょうか。画面上に現れるオブジェクトを操作するには，オブジェクト構造体の中の各データをいじってやればいいわけですね。で，シューティング・ゲームに登場するオブジェクトは，表示するだけなら，前述もので十分なのですけれど，実際にゲームに昇華するにはまだまだ情報が足りません。

たとえばこのオブジェクトはキーで動くのか，それともコースを移動するのか，それともランダムに移動するのか。たとえばこのオブジェクトは敵なのか味方なのか。たとえばこのオブジェクトは何度敵や弾と衝突することができるのか。

考えていけば必要な情報はいくつも出てくることでしょう。それがゲーム性に強く現れるわけですが，ここまでくるとプログラマーがそれぞれ考えてください，と言うしかありません。

ですが，前述の情報はすべて，個々のオブジェクトにあるべきものです。自機であろうが弾であろうが，移動方法や衝突回数…つまりヒットポイントのようなものですが，これらはどのオブジェクトでも使える情報と言えるでしょう。このような情報は，描画のための情報と同様に，OBJDATA構造体に含めてしまえばいいのです。

で，自機としたいオブジェクトの「移動方法タイプ」に，あらかじめ「キーで動く」という値を入れておきます。プログラム的に言えば，

```
pObjData[0].dwMoveType = 0;
```

という感じにしてはどうでしょう。dwMoveTypeというのが追加した「移動方法タイプ」を保存しておく場所で，「0」が「キーで動く」ことを示した値とします。ここには，キーで動かず，プログラムによる移動の場合はその移動ルートや方法を示した番号を入れておくことにしましょう。そうすることで，さまざまな移動を1か所にまとめて記述することができ，データ管理もプログラム保守も非常に簡単になります。

計算部分のプログラムは，**リスト2**のような感じで組むことになります。

さきほど例に出した「このオブジェクトは敵なのか味方なのか」や「このオブジェクトは何度敵や弾と衝突することができるのか」という情報も同様にOBJDATA構造体内に含めてしまいます。前者は，弾と敵の衝突判定時に使えますし，後者は実際に当たったとき，そこから爆発効果に移行するのか，それともそのまま向かってくるのかというゲーム性に使うことができます。

このような感じで核となる部分を作っておけば，ゲーム性を追加したり変更したりというときに，必要な情報をOBJDATA構造体を操作するだけでよいので，プログラムも非常に作りやすいでしょう。

次回はこのオブジェクト的考えかたをもっと複雑に絡めたプログラムの組みかたを紹介したいと思います。

《リスト2》計算部分のリスト

```
for ( i = 0; i < 10; i++ ) {
    // 移動処理
    switch( pObjData[i].dwMoveType ) {       case 0:
        {   // キーによる移動
            break;
        }
        case 1:
        {   // プログラムによる移動1
            break;
        }
        case 2:
        {   // プログラムによる移動2
            break;
        }
            :
            :
    }
}
```

catlike最新情報

お待ちかね。「いつか，何処かで。」の一般公開がスタートしました。長ーくお待たせしてしまい，ご心配をおかけいたしました。でもでも，お待たせしたかいあって，ちょっとよさげな作品になりましたので，ぜひ見てくださいね。まずは第１話から，ご覧くださいませ！





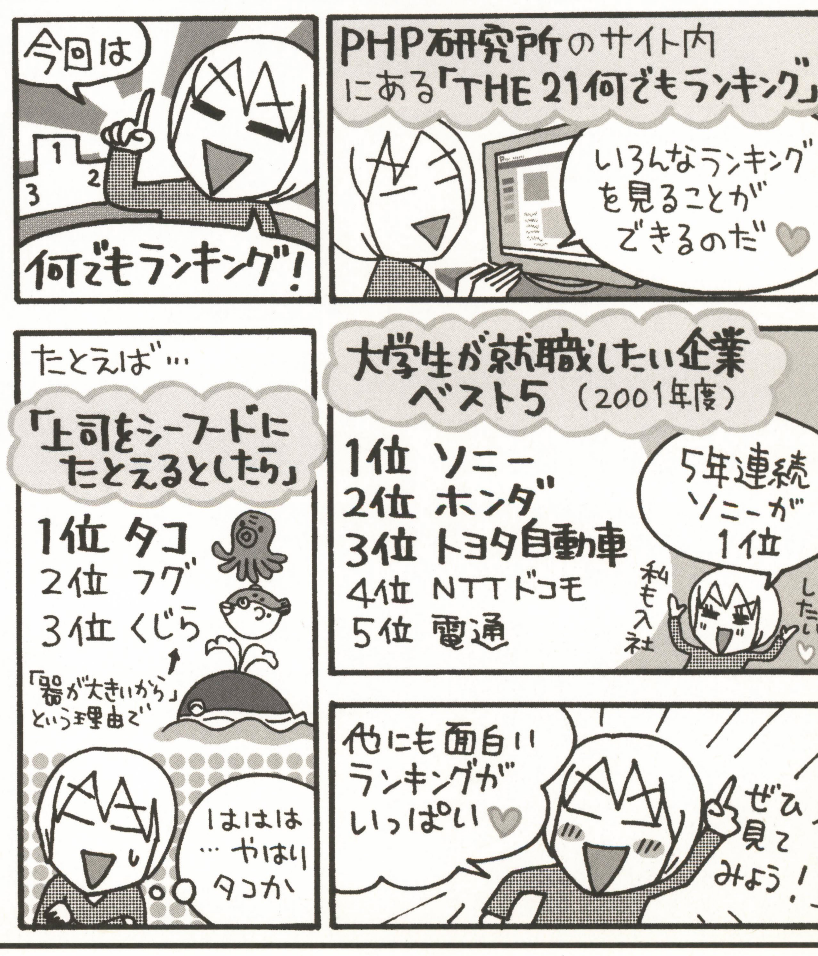
今回は突然に「いろんなモノのランキングが知りたい」と思い立ちYAHOO!で検索してみたところ発見！PHP研究所のサイト内にある「THE21なんでもランキング」というコーナーです。

いろんな企業や政府でとったアンケート結果をここでまとめて見ることができるのですが，けっこう思いもよらない結果が出ていたりして驚きます。

たとえば，子供が仲良しと思う家族ランキング第1位がお母さん・・そして2位はなんとペット！　で，お父さんはというと5位！？男子の1位はお父さんだったのですが，女子との総合になると順位を落としてしまいこういう結果になったのだとか。

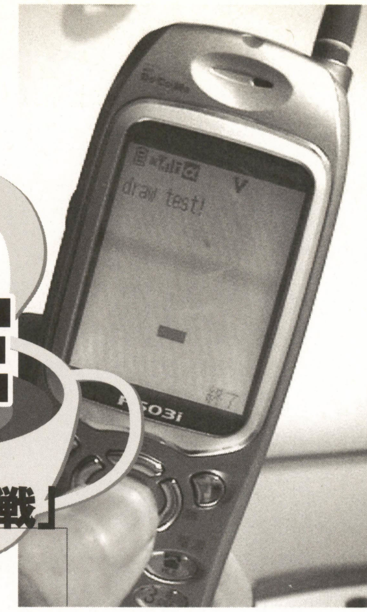
この他にもBOOK／ビデオ／CDのランキング…口コミプロバイダ情報等，いろんな気になる情報を入手できるので，ぜひチェックしてみてください。

PHP INTERFACE「THE21 なんでもランキング」
http://www.php.co.jp/fun/the21/index.html



携帯電話で動くゲームを作ろう!!

君にもできる「iアプリ」制作



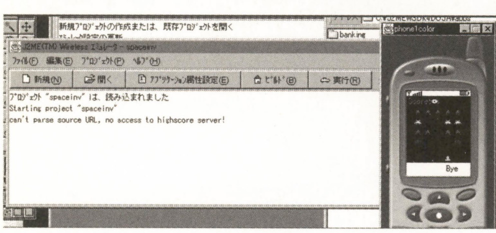
第6弾「iアプリでロールプレイング・ゲームに挑戦」

動作の遅いiアプリには非リアルタイムのゲームが向いているのではないかと思い，ロールプレイング・ゲームを作ってみました。（松原拓也）

iアプリの作りかた

①サンマイクロシステムズのWEBサイト（http://java.sun.com/）からJava2の開発環境「Java 2 SDK Standard Edition 1.3.0」をダウンロードします。インストールすると“C:¥jdk1.3”というディレクトリにJavaコンパイラが展開されます。

②NTTドコモのWEBサイト（http://www.nttdocomo.co.jp/i/java.html）からiアプリの開発ツール「J2ME Wireless SDK for the DoJa release2.2（以下，DoJa）」をダウンロードします。ダウンロードには名前の登録が必要です。インストールすると“C:¥J2MEWSDK4DOJA”フォルダに主要なファイルが展開されます。



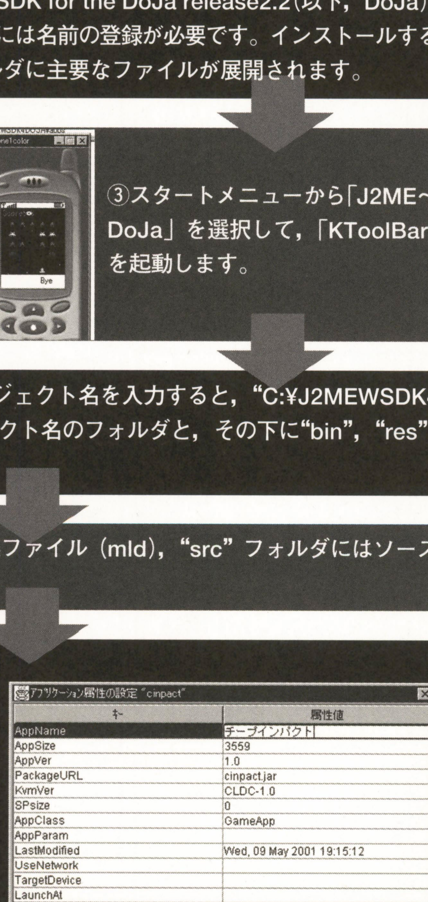
③スタートメニューから「J2ME～DoJa」を選択して，「KToolBar」を起動します。

④「新規」ボタンを押して，新規プロジェクト名を入力すると，“C:¥J2MEWSDK4DOJA¥apps”フォルダの下にプロジェクト名のフォルダと，その下に“bin”，“res”，“src”フォルダが作成されます。

⑤“res”フォルダに画像（gif）や音楽ファイル（mld），“src”フォルダにはソースファイル（java）を格納します。

⑥「アプリケーション属性設定」ボタンを押して，キーの値を入力します（“AppSize”と“Last Modified”は自動的に更新されます）。

AppName =（アプリケーション名）
AppVer = 1.0
AppClass =（最初に実行するクラス名）
SPsize =（スクラッチパッドのサイズ）
KvmVer = CLDC-1.0

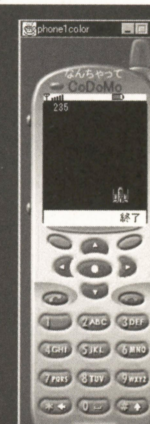


アプリケーション属性の設定 “cinpact”

キー	属性値
AppName	チープインパクト
AppSize	3559
AppVer	1.0
PackageURL	cinpact.jar
KvmVer	CLDC-1.0
SPsize	0
AppClass	GameApp
AppParam	
LastModified	Wed, 09 May 2001 19:15:12
UseNetwork	
TargetDevice	
LaunchAt	

⑦「ビルド」ボタンを押すと，コンパイルとプリベリファイが一括して行われ，jarファイルとjamファイルが“bin”フォルダに生成されます。jarファイルは“res”フォルダの中身とクラスファイルがzip形式で圧縮されたものです。

⑧エディタで“C:¥J2MEWSDK4DOJA¥lib¥devices¥phone1 color”フォルダの“phone1 color.properties”を以下のように書き換えます（値は120×130ピクセル時の設定です）。書き換えたら「編集－設定」をメニュー選択して，「デバイス指定」を“phone1 color”にします。



```
  :
screen.x=27
screen.y=65
  :
screen.width=120
screen.height=156
  :
icon.up: 44, 138, off
  :
icon.down: 44, 146, off
  :
screenPaintableRegion.width=120
screenPaintableRegion.height=130
  :
softbutton.0=0,138,36,18, softButton
softbutton.1=90,138,36,18, softButton
  :
```

⑨「実行」ボタンを押すと，携帯電話エミュレータ上でiアプリが実行されます。

i アプリでロールプレイング・ゲームに挑戦

『スライム・クエスト』



▲《写真1》移動中の画面。中心が主人公。下に見えるのが宝石

ゲーム内容

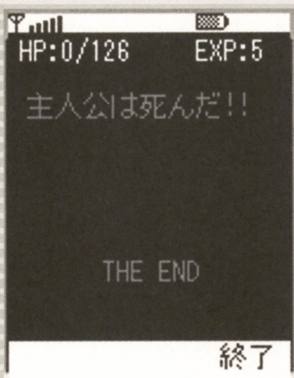
ものすごくシンプルなロールプレイング·ゲームです。方向キーで移動して，宝石のある場所に向かってください（**写真1**）。敵と遭遇すると戦闘シーンに突入します。戦闘ではいずれかのボタンを押してください。主人公と敵の攻撃が乱数で行われ，結果が表示されます（逃げることはできません）。戦闘中，主人公の体力（HP）がなくなるとゲーム・オーバーです（**写真2**）。

体力は歩くことで回復します。敵を倒すと，経験値（EXP）が増えます。経験値が増えると主人公の体力が上がります（**写真3**）。敵はすべてスライムです。スライムの外見はまったく変化がありませんが，南に向かえば向かうほど極端に強くなります。ですので，最初は北の草原で主人公のレベルをアップさせましょう。

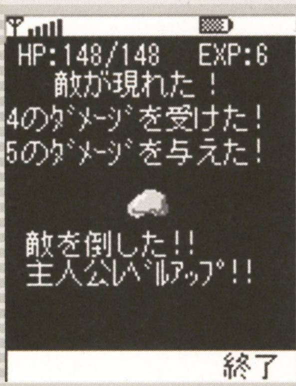
宝石を手に入れるとゲームクリアになります（**写真4**）。

入力方法

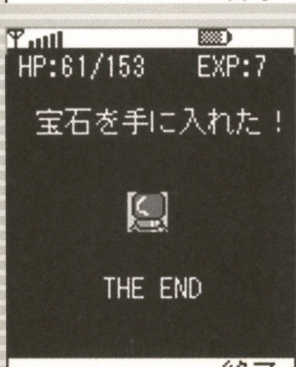
リスト1～4をエディタなどで入力してください。さらにグラフィックツールなどで，「芝生」「岩」「主人公」「宝石」「海」「敵」のGIF画像を描き，“0.gif”～“5.gif”のファイル名で保存してください（**写真5**）。



◀《写真2》体力のダメージが大きくなりすぎるとゲーム・オーバー



◀《写真3》キーを押すだけで進展する戦闘シーン。敵を倒して経験値を稼ぐと体力が上がる



◀《写真4》宝石を手に入れるとゲームクリア！このゲーム，謎解きとかはとくにありません……

《リスト1》JAM ファイル “rpg.jam”

```
PackageURL = rpg.jar
AppSize = 3976
AppName = RPG
AppVer =
AppParam =
AppClass = GameApp
SPsize = 0
LastModified = Mon, 20 Aug 2001 12:13:32
KvmVer = CLDC-1.0
```

《リスト2》ソースファイル “GameApp.java”

```
/*------------------------------------------------------------*/
/* "GameApp.java"  by 松原拓也*/

import com.nttdocomo.util.*;  /*クラスのインポート*/
import com.nttdocomo.ui.*;

public class GameApp extends IApplication
{
    public void start()
    {
        GameScreen screen = new GameScreen(); /*キャンバスの作成*/
        Display.setCurrent(screen);  /*キャンバスのカレント表示*/
    }
}
```

《リスト3》html ファイル “rpg.html”

```
<HTML>
<HEAD>
<TITLE>i-appli</TITLE>
</HEAD>
<BODY>
 i アプリゲーム<BR>
<OBJECT declare id="b" data="rpg.jam" type="application/x-jam">
RPG</OBJECT><BR>
<A ijam="#b" href="rpg.html">RPG</A>
<br>
</BODY>
</HTML>
```

そして，「Doja」などのｉアプリ開発環境でビルドします。ビルドに成功すると，“rpg.jar”というJARファイルが生成されます。実行はJARファイル，JAMファイル，HTMLファイルをサーバにアップロードしてから，iアプリ対応携帯電話や携帯電話エミュレータで呼び出します（前ページの「ｉアプリ開発方法」を参考にしてください）。

技術的な解説

❶マップデータは“iMap”という2次元配列で管理しています（**写真6**）。その値はキャラクターのIDに対応しています。あと，配列の外を読み込んでしまわないよう，マップは3個分の「海」タイルで囲んでいます。

❷プログラムの簡略化のため，GIF画像をforループで連続して読み込みます。読み込んだイメージはImageの1次元配列に格納されます。

❸主人公の歩行用サブ・ルーチンです。主人公は草原の上しか歩けませんので，前進する前に地面のチェックを行います。さらに，敵と遭遇するかを乱数で設定します。敵と遭遇する確率は2割くらいです。

❹ペイントメソッドです。ここで戦闘シーンと移動シーンとで処理を分岐しています。ただ，戦闘中の処理をすべて入れてしまい，ソースがちょっと見づらくなってしまいました。できるだけ，キー入力側で処理したほうがいいと思います。

❺主人公の存在する座標は“iX,iY”です。この座標を中心として7×7個のタイルを画面に表示します。

❻今回は非リアルタイム・ゲームなので，スレッド（Thread）を使いませんでした。そのため，キー入力があったときだけ処理を行っています。キーが押されたときには“processEvent”というメソッドが呼ばれ，パラメータの値が“Display.KEY_PRESSED_

《リスト4》ソースファイル“GameScreen.java”

```
/*------------------------------------------------------------*/
/* 「RPG」キャンバス "GameScreen.java"   by 松原拓也 */

/* クラスをインポート*/
import com.nttdocomo.ui.*;
import com.nttdocomo.io.*;
import javax.microedition.io.*;
import java.util.*;
import java.io.*;

public class GameScreen extends Canvas {

    final int TIPW = 17;      /* キャラの幅・高さ[ピクセル] */
    final int PATMAX = 6;     /* 総パターン数*/
    final int MAPW  = 7;               /*ピンホールのタイル数*/
    final int MAPH  = (MAPW-1)/2;   /*中心からのタイル数*/
    final int SEQ_BADEND=3;  /*シーケンス：ゲームオーバー*/
    final int SEQ_END=2;     /*シーケンス：エンディング*/
    final int SEQ_FIGHT=1;   /*シーケンス：戦闘中*/
    final int SEQ_WALK=0;    /*シーケンス：移動中*/
    final int ID_NONE=0;    /*キャラID：何も無し*/
    final int ID_HOSEKI=3;  /*キャラID：宝石*/
    final int ID_TEKI=5;    /*キャラID：敵*/
    final int ID_MY=2;      /*キャラID：主人公*/

    Image imgStar[] = new Image[PATMAX];     /* イメージ配列 */

    int iMap[][]={   /*マップデータ*/   ◀············❶
        {4,4,4,4,4,4,4,4,4,4,4,4,4,4,4,4,4,4,4,4},
        {4,4,4,4,4,4,4,4,4,4,4,4,4,4,4,4,4,4,4,4},
        {4,4,4,4,4,4,4,4,4,4,4,4,4,4,4,4,4,4,4,4},
        {4,4,4,0,0,1,1,0,0,0,0,0,0,0,0,0,4,4,4},
        {4,4,4,1,0,0,0,0,0,0,1,1,1,0,0,1,0,4,4,4},
        {4,4,4,0,1,1,0,0,1,0,1,0,0,0,0,0,4,4,4},
        {4,4,4,0,0,0,0,0,0,1,1,1,0,1,1,0,4,4,4},
        {4,4,4,0,0,1,0,1,0,0,0,0,0,0,0,0,4,4,4},
        {4,4,4,0,0,0,0,0,0,1,1,0,0,0,1,0,0,4,4,4},
        {4,4,4,1,0,1,1,1,1,0,0,1,0,0,0,0,0,4,4,4},
        {4,4,4,1,0,0,1,0,0,0,1,0,1,0,1,1,1,4,4,4},
        {4,4,4,0,0,0,0,0,0,1,1,0,1,1,1,3,0,4,4,4},
        {4,4,4,0,0,1,1,0,0,0,0,0,0,0,0,0,1,4,4,4},
        {4,4,4,4,4,4,4,4,4,4,4,4,4,4,4,4,4,4,4,4},
        {4,4,4,4,4,4,4,4,4,4,4,4,4,4,4,4,4,4,4,4},
        {4,4,4,4,4,4,4,4,4,4,4,4,4,4,4,4,4,4,4,4},
    };

    int iX=3 ,iY=3;         /*主人公座標*/
    int iSeq=SEQ_WALK;          /*シーケンス*/
    int iHp=100,iDmg=0,iExp=0;   /*体力、ダメージ、経験値*/
    int iTp;
    Random rnd = new Random();   /* 新しい乱数列を作成*/

    public GameScreen() {    /*コンストラクタ*/
        int i;
        MediaImage mi;

        setSoftLabel(Frame.SOFT_KEY_2, "終了");

        for(i=0 ;i < PATMAX ;i++) {    ◀············❷
            mi = MediaManager.getImage("resource:///"+ i +".gif");
            try {
                    mi.use(); /* イメージの読み込み */
            } catch (ConnectionException e) {
            }
            imgStar[i] = mi.getImage(); /* GIF画像読込 */
        }
    }

    /*歩行処理サブルーチン*/
    public void subWalk(int iX1,int iY1) {  ◀············❸
        int iChr;

        iDmg -= Math.abs(rnd.nextInt() % 10);
        if(iDmg < 0)
            iDmg=0; /*ダメージ回復*/

        iChr = iMap[iY+iY1][iX+iX1];
        if(iChr == ID_NONE) {
            iX += iX1;
            iY += iY1;
        }
        if( Math.abs(rnd.nextInt() % 100) > 80) {
            iTp = Math.abs(rnd.nextInt() % 10)+(iY*20); /*敵体力*/
            iSeq = SEQ_FIGHT;    /*エンカウント!!*/
        }
        if(iChr == ID_HOSEKI) {
            iSeq = SEQ_END;    /*エンディング*/
        }
    }
```

```
public void paint(Graphics g) {  /*ペイントメソッド*/   ◀·····························❹
     int iX1, iY1, iD1, iT1;
     int     iChr;

     g.lock();  /* 画面のロック */
     g.setColor(g.getColorOfRGB(0, 0, 0));
     g.fillRect(0, 0, getWidth(), getHeight() ); /*画面クリア*/

     switch(iSeq) {
     case SEQ_BADEND: /*ゲームオーバー*/
         g.setColor(g.getColorOfRGB(255, 0, 255));
         g.drawString("主人公は死んだ!!" ,8 ,35 );
         g.drawString("THE END" ,40 ,105 );
         break;
     case SEQ_END: /*エンディング*/
         g.setColor(g.getColorOfRGB(255, 255, 0));
         g.drawString("宝石を手に入れた！" ,12 ,35 );
         g.drawString("THE END" ,40 ,105 );
         g.drawImage(imgStar[3] ,50 ,60 );
         break;
     case SEQ_FIGHT: /*戦闘シーン*/
         g.setColor(g.getColorOfRGB(255, 255, 255));
         g.drawString("敵が現れた！" ,20 ,24 );
         g.drawImage(imgStar[ID_TEKI] ,50 ,60 );

         iD1 = Math.abs(rnd.nextInt() % 5);
         g.drawString(iD1+"のﾀﾞﾒｰｼﾞを受けた!" ,0 ,38 );
         iDmg += iD1;

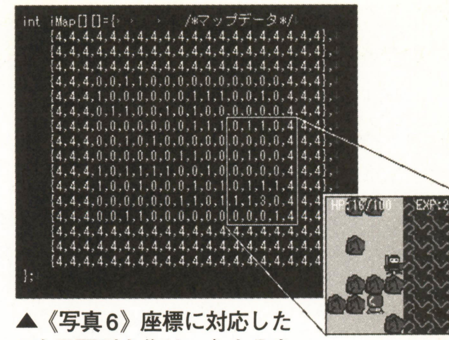
         iT1 = Math.abs(rnd.nextInt() % 10);
         g.drawString(iT1+"のﾀﾞﾒｰｼﾞを与えた!" ,0 ,52 );
         iTp -= iT1;
         if(iTp <= 0) {
             g.drawString("敵を倒した!!" ,8 ,90 );
             iExp ++;
             if((iExp % 3)==0) {
                 g.drawString("主人公ﾚﾍﾞﾙｱｯﾌﾟ!!" ,8 ,102 );
                 iHp += (Math.abs(rnd.nextInt() % 10)+20);
                 iDmg=0;
             }
             iSeq = SEQ_WALK;
         }
         if((iHp-iDmg) <= 0)
              iSeq = SEQ_BADEND;  /*死亡*/

         break;
     case SEQ_WALK:              /*歩行シーケンス*/   ◀·························❺
         for(iY1=0 ;iY1<MAPW ;iY1++) {
             for(iX1=0 ;iX1<MAPW ;iX1++) {
                 iChr = iMap[iY+iY1-MAPH][iX+iX1-MAPH];
                 g.drawImage(imgStar[iChr] ,iX1*TIPW ,iY1*TIPW );
             }
         }
         g.drawImage(imgStar[ID_MY] ,MAPH*TIPW ,MAPH*TIPW );     /*主人公*/
         break;
     }
     g.setColor(g.getColorOfRGB(255, 255, 255));
     g.drawString("HP:"+(iHp - iDmg)+"/"+iHp , 5, 12);
     g.drawString("EXP:"+iExp, 80, 12);

     g.unlock(false);  /* ロック解除 */
}

public void processEvent(int nType, int nParm)   ◀·······························❻
{
     if(nType == Display.KEY_PRESSED_EVENT)
     {
         if(nParm == Display.KEY_SOFT2)
             IApplication.getCurrentApp().terminate();

         if(iSeq==SEQ_WALK) {
             switch(nParm)
             {   /*移動処理*/
             case Display.KEY_DOWN:
                 subWalk( 0 ,1);
                 break;
             case Display.KEY_LEFT:
                 subWalk(-1 ,0);
                 break;
             case Display.KEY_RIGHT:
                 subWalk( 1 ,0);
                 break;
             case Display.KEY_UP:
                 subWalk( 0 ,-1);
                 break;
             }
         }
         repaint(); // 再描画する。
     }
}
}
```



▲《写真6》座標に対応した2次元配列を作り，主人公を中心としてキャラクターのデータを取り出す

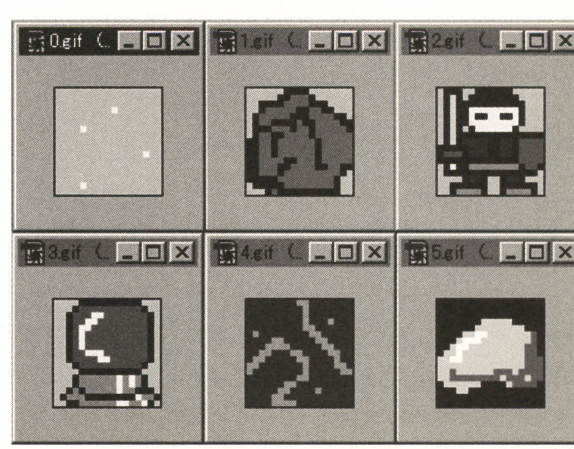
EVENT”を通知します。

最後に

レベルアップを続けていれば敵を倒せるようになる，というゲーム・システムが安易すぎました(反省)。やはり戦闘シーンは攻撃方法を選べたり，敵に弱点があったりと，戦略性がほしいところです。

この作品のJARファイルの大きさは3.8Kバイトでした。現在，ｉアプリのサイズの上限は10Kバイトまでですから，内容をもっと積め込めそうです。敵やパラメータを増やしてみたり，フラグを導入してみたり，サーバからマップデータを受信できるようにしたりすると本当におもしろいゲームに変身するかもしれません。

＊本プログラムのソースはベーマガWEBサイトからダウンロードすることができます。さらにｉアプリ対応の携帯電話で実行することができます。
http://www.basicmagazine.net/



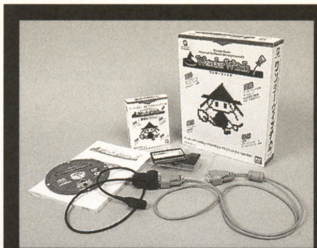
▲《写真5》17×17ピクセルのGIF画像を6枚作成する



チャレンジ!! ワンダーウィッチ

ワンダースワン ホビープログラミングツール Wonder Witch ワンダーウィッチ

Challenge WonderWitch



ワンダースワン・ホビープログラミングツール
『ワンダーウィッチ（WonderWitch）』
定価16,800円（税別）

◆パッケージ内容
・PC-WonderSwan 接続ケーブル
（スワンケーブル，クロスケーブル）
・ワンダーウィッチ専用カートリッジ
・CD-ROM　・マニュアル

◆動作環境
機種: PC/AT 互換機
OS: Windows 98／Me／NT4.0／2000
CPU: Pentium 133MHz以上
メモリ: 32MB以上（推奨64MB以上）
ディスプレイ: 640×480，256色以上
シリアルポート: 9600/38400bps
＊専用カートリッジのみ単体の購入も可能。
3,980円（税別）。

◆製品について問い合わせ
wonderwitch-info@bandai.co.jp

◆通信販売について問い合わせ
netshop@bandai.co.jp

第12回「ライン描画に挑戦」

ワンダーウィッチにはBASICの「Line」文のような，画面に線を描くライブラリが用意されていません。そこで，今回はライン描画のための関数を自作して，それを使ったゲームを作ってみたいと思います。（松原拓也）

線を描くアルゴリズム

では，二つの点（X1,Y1）と（X2,Y2）の間に直線を描きたいと思います。一体，どういう処理をすればいいでしょう。

最初に思いつくのは，数学で習った「一次関数」を使うことです。X座標の差をxd，Y座標の差をydとすると，傾きは

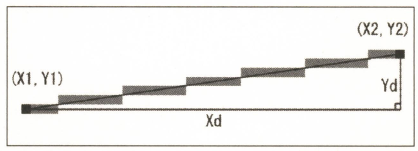
a=yd/xd

となります。X座標あたりのY座標を「Y=aX+b」の式で求めて，(X,Y）の点を描いていけばいいはずです。

しかし，もっと処理が高速な「ブレセンハム（Bresenham）の線分発生」というアルゴリズムがありますので，それを使ってみましょう（第1図）。以下が「ブレセンハム～」の処理手順です。なお，この処理は傾きが1以下（yd < xd）という条件になります。

①eに0を代入
②xを1ピクセル進める
③eに傾きaを加える
④もしe>=0.5ならyを1ピクセル進め，eから1を引く
⑤点(x,y)を描いたら②に戻る

このアルゴリズムのポイントは，「xとyが等しく斜めに進むには傾き=1である」ということを応用している点です。このため，傾きを累積するためにeという変数が用意され，傾きが0.5以上（四捨五入のため）ならXと一緒にYも1ピクセルずつ進められます（このとき，eから傾き1が引かれます）。



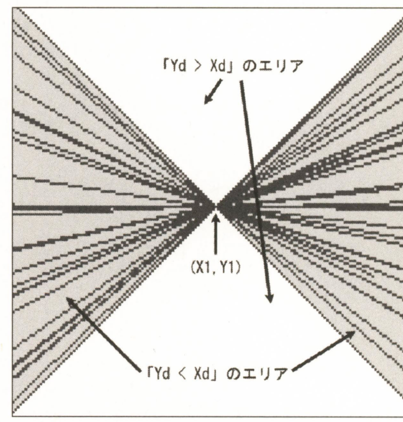
▲《第1図》「ブレゼンハムの線分発生アルゴリズム」で描いた直線の拡大図

そして，このeにxdを掛け，さらに2倍すると以下のように式を展開することができます。

①eに0を代入
②xを1ピクセル進める
③eにyd＊2を加える
④もしe>=xdならyを1ピクセル進め，eからxd＊2を引く
⑤点(x,y)を描いたら②に戻る

xdとydは整数ですから，これでなんと少数を使わずに直線を描けるようになりました。あと，(yd > xd）の場合は，「傾き1以下」という条件が成り立ちませんので，xとyの関係を入れ替えて計算します（第2図）。

なお，このアルゴリズムはシューティング・ゲームで敵弾の追尾にも応用できます。

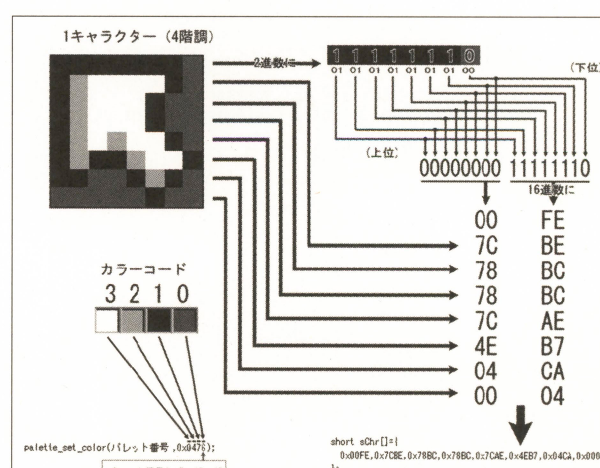


▲《第2図》傾き1以下の範囲

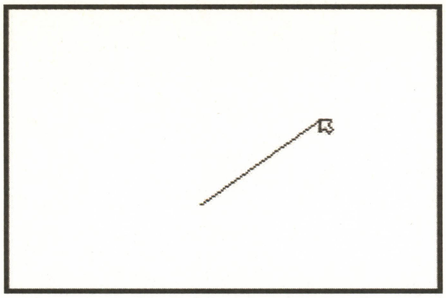
スクリーン描画方法

ワンダーウィッチでは，点や線はスクリーンというキャラクターが並んだバックグラウンドに描かれます。

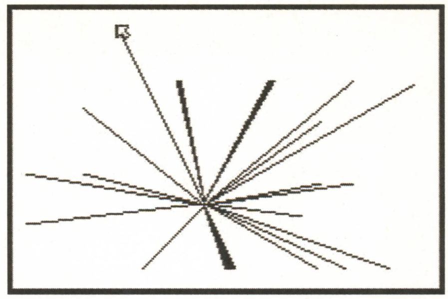
まずスクリーンの初期化です。



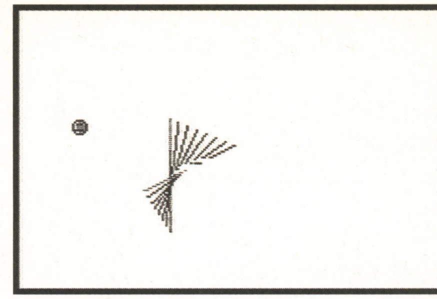
▲《第3図》ワンダーウィッチのキャラクター・データ・フォーマットのサンプル



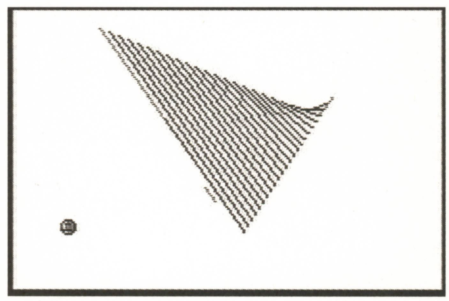
▲《画面1》線を描くだけのサンプル・プログラムを実行中



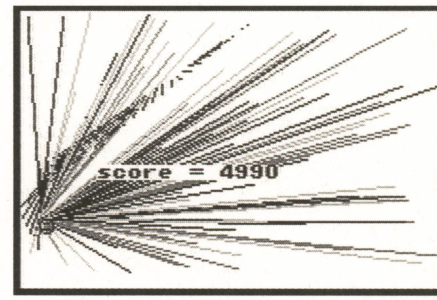
▲《画面2》あらゆる方向に線が引けることを確認



▲《画面3》左側の丸いのが主人公。中央の線のかたまりが敵。敵はクネクネと画面内をバウンドする



▲《画面4》敵はだんだんと巨大化していく。主人公ピンチ



▲《画面5》敵に接触すると大爆発を起こしてゲーム・オーバー!!

「text_window_init」関数で，キャラクターをスクリーン上に敷き詰めます（28×18個)。そして，（X,Y)に点を描きたい場合，スクリーンの（X/8,Y/8）にあるキャラクター番号を調べ，そのキャラクター・データを更新します。

キャラクター・データの取得は「font_get_data」関数を使い，登録は「font_set_colordata」関数を使います。キャラクター・データは一つあたり8×8ピクセル単位，16バイトで扱います。キャラクター・データのフォーマットは**第3図**を参考にしてください。

そして，線を描画するだけの簡単なサンプルを作ってみました（**画面1**)。成功です（**画面2**)。この関数を応用してゲームを作ってみたいと思います。

ライン描画を使ったゲーム

方向キーで主人公を動かし，敵をかわす時間を競うゲームです（**リスト1**)。

敵はタイトーの「Qix」に出てくるクイックス(「南京玉すだれ」みたいなもの)をイメージしてみました（**画面3**)。画面内をバウンドし続け，時間と共に巨大化します（**画面4**)。

主人公がこの敵に触れるとゲーム・オーバーです。ゲーム・オーバーになるとスコアが表示されます（**画面5**)。

技術的な解説

敵は，今回自作した「subLine」関数を使って描いてみました。「subLine」関数は処理に負担がかかるようで，線が長くなると処理落ちしてしまいました。どうにか高速化したいと思ったのですが，今回は諦めました。

ゲーム・オーバー時の大爆発も「subLine」関数を使っています。当たり判定のために，指定した座標のカラーコードがわかる「fncPoint」という関数を自作してみました。

まとめ

このライン描画関数を使って，ベクタスキャン風のゲームや昔のアドベンチャー・ゲームっぽいゲームを作ってみるとおもしろいかもしれません。

「ワンダーゲート」プレゼント発表

7～8月の間「投稿プログラムコーナー」にワンダーウィッチ用のプログラムを投稿してくれた人の中から厳正な抽選の結果，中野真良さんに決定しました。おめでとうございます。

《リスト1》『ライン描画を使ったゲーム』のプログラム・リスト

```
/*-----------------ライン描画を使ったゲーム     by 松原 */
#include <sys/bios.h>
#include <stdio.h>
#include <stdlib.h>


#define CHR_PIX    8    /* 1キャラクターの1辺の長さ[ピクセル]*/
#define CHR_BYTE 16    /* 1キャラクターのデータ数[バイト](4階調時) */

typedef struct {
   int x1;
   int y1;
   int x2;
   int y2;
} ST_TAMA;   /*敵（南京玉すだれ）用構造体*/

#define TAMAMAX 30       /*敵の最大の長さ*/
ST_TAMA tama[TAMAMAX];   /*敵の座標バッファ*/

char cBuffer[CHR_BYTE];     /*PSET用キャラクターバッファ*/

short sChr[]={
  /*主人公キャラクターデータ*/
  0x003C,0x3C62,0x56E9,0x7EC1,0x7EC1,0x02FD,0x3C62,0x003C
};

#define SPN_MIN  0      /*スプライト番号：最小*/
#define SPN_MY   0      /*スプライト番号：カーソル*/
#define SPN_MAX  1      /*使用スプライト数*/
```

```
#define CHN_MY    511    /*キャラクター番号：主人公*/

#define CPN_TEXT 0    /*キャラクターパレット番号：スクリーン用*/
#define CPN_SP    8    /*キャラクターパレット番号：スプライト有効開始値(常数)*/
#define CPN_MAS  12    /*キャラクターパレット番号：透明色有効(常数)*/
#define CFM_CPN ((CPN_MAS-CPN_SP)<< CFSFT_PALETTE)  /*パレット番号割り振り*/


/*-------絶対値を求める関数----*/
/*引数（絶対値を求めたい値）*/
int fncAbs(int d)
{
  if(d < 0)
    return(-d);
  else
    return(d);
}

/*-------------符号を求める関数----*/
/*引数（符号を求めたい値）*/
int fncSgn(int d1)
{
  if(d1 > 0)
    return(1);  /*プラス*/
  else
    return(-1); /*マイナス*/
}

/*----------------------------------*/
/*ドット情報取得（x座標 ,y座標 ）*/
/*戻り値：カラーコード0～3*/
char fncPoint(int x ,int y)
{
  int iChn;
  int iOfst;
  char cMask;
  char color=0;

  iChn=screen_get_char1(SCREEN1 ,x/CHR_PIX ,y/CHR_PIX) & 0x1FF;
  font_get_data(iChn ,1 ,cBuffer);
  iOfst = (y % CHR_PIX) * 2;
  cMask = 0x80 >> (x % CHR_PIX);

  if(cBuffer[iOfst] & cMask)
    color = 1;

  iOfst++;
  if(cBuffer[iOfst] & cMask)
    color |= 2;

  return(color);
}

/*----------------------------------*/
/*ドット描画（x座標 ,y座標 ,カラーコード0～3 ）*/
void subPset(int x ,int y ,char color)
{
  int iChn;  /*キャラクター番号*/
  int iOfst; /*オフセット*/
  char cMask; /*マスク*/

  iChn=screen_get_char1(SCREEN1 ,x/CHR_PIX ,y/CHR_PIX) & 0x1FF;
  font_get_data(iChn ,1 ,cBuffer);   /*キャラクターデータ取得*/
  iOfst = (y % CHR_PIX) * 2;
  cMask = 0x80 >> (x % CHR_PIX);

  cBuffer[iOfst] &= ~cMask;
  cBuffer[iOfst] |= cMask * (color & 1);
  iOfst++;
  cBuffer[iOfst] &= ~cMask;
  cBuffer[iOfst] |= cMask * ((color & 2) >> 1);

  font_set_colordata(iChn ,1 ,cBuffer);   /*キャラクター定義*/
}

/*----------------------------------*/
/*ライン描画（x1座標 ,y1座標 ,x2座標 ,y2座標 ,カラーコード ）*/
void subLine(int x1 ,int y1 ,int x2 ,int y2 ,char color)
{
  int xd,yd,xs,ys,e;

  xd = fncAbs(x2 - x1);  /* X座標の距離*/
  yd = fncAbs(y2 - y1);  /* Y座標の距離*/
  xs = fncSgn(x2 - x1);  /* X方向の移動量(1ピクセル分)*/
  ys = fncSgn(y2 - y1);  /* Y方向の移動量(1ピクセル分)*/

  if( yd < xd ) {
    e = 0;
    while( x1 != x2) {
      subPset (x1, y1 ,color); /*ドット描画*/
      x1 += xs;
      e += (2 * yd);
      if(e >= xd) {
        y1 += ys;
        e -= (2 * xd);
      }
    }
  }else{
    e = 0;
    while( y1 != y2) {
      subPset (x1, y1 ,color); /*ドット描画*/
      y1 += ys;
      e += (2 * xd);
      if(e >= yd) {
        x1 += xs;
        e -= (2 * yd);
      }
    }
  }
}


/*--------------------メインプログラム*/
```

```
void main () {
  #define MOVEPIX  3  /*自分キャラの移動量*/
  int iKey,i;
  char sBuf[28];
  int iTmax=5;    /*敵の長さ*/
  int iTx1a=0,iTy1a=1;  /*敵移動量*/
  int iTx2a=0,iTy2a=1;
  int iMx=(LCD_PIXEL_WIDTH/2); /*主人公座標*/
  int iMy=(LCD_PIXEL_HEIGHT/2);
  unsigned long ulTback,ulTstart;

  lcd_set_color(0x6420, 0xfca8);   /* LCD階調データ初期化 */

  text_set_screen(SCREEN1);   /*スクリーン初期化*/
  text_set_palette(CPN_TEXT);
  text_window_init(0,     0,     TEXT_SCREEN_WIDTH,
TEXT_SCREEN_HEIGHT, 0);
  palette_set_color(CPN_TEXT ,0x7420);/*キャラクターパレット
設定*/

  font_set_colordata(CHN_MY  ,(sizeof(sChr)  /  CHR_BYTE)
,sChr);
  sprite_set_range(SPN_MIN ,SPN_MAX );/*スプライト範囲を設定
*/
  display_control( DCM_SPR | DCM_SCR1);   /*スクリーン&スプ
ライト表示設定*/
  palette_set_color(CPN_MAS ,0x0470); /*キャラクターパレット
設定*/

  sprite_set_char(SPN_MY ,CFM_CPN  |  CHN_MY);   /*主人公スプ
ライト設定*/

  ulTstart=sys_get_tick_count();
  ulTback=ulTstart;

  while (1) {
    if((iKey = key_press_check())== KEY_START)
      break;

    /*------主人公移動*/
    if((iKey & KEY_DOWN1)&&(iMy < (LCD_PIXEL_HEIGHT-1)))
      iMy += MOVEPIX;

    if((iKey & KEY_UP1)&&(iMy > 0))
      iMy -= MOVEPIX;

    if((iKey & KEY_RIGHT1)&&(iMx < (LCD_PIXEL_WIDTH-1)))
      iMx += MOVEPIX;

    if((iKey & KEY_LEFT1)&&(iMx > 0))
      iMx -= MOVEPIX;

    sprite_set_location(SPN_MY ,iMx ,iMy);

    /*---------敵移動-----*/
    i=iTmax;
    subLine(tama[i].x1    ,tama[i].y1    ,tama[i].x2
,tama[i].y2 ,0);

    for(i=iTmax ;i>=1 ;i--) {
      tama[i].x1 = tama[i-1].x1;  /*敵座標をバッファ*/
      tama[i].y1 = tama[i-1].y1;
      tama[i].x2 = tama[i-1].x2;
      tama[i].y2 = tama[i-1].y2;
    }

    tama[0].x1 += iTx1a;
    tama[0].y1 += iTy1a;
    tama[0].x2 += iTx2a;
    tama[0].y2 += iTy2a;
    subLine(tama[0].x1    ,tama[0].y1    ,tama[0].x2
,tama[0].y2 ,3);

    /*------敵のバウンド---*/
    if(tama[0].x1 <= 0)
      iTx1a = (rand() % 4)+2;
    else if(tama[0].x1 >= LCD_PIXEL_WIDTH)
      iTx1a = -(rand() % 4)-2;

    if(tama[0].y1 <= 0)
      iTy1a = (rand() % 4)+2;
    else if(tama[0].y1 >= LCD_PIXEL_HEIGHT)
      iTy1a = -(rand() % 4)-2;

    if(tama[0].x2 <= 0)
      iTx2a = (rand() % 4)+2;
    else if(tama[0].x2 >= LCD_PIXEL_WIDTH)
      iTx2a = -(rand() % 4)-2;

    if(tama[0].y2 <= 0)
      iTy2a = (rand() % 4)+2;
    else if(tama[0].y2 >= LCD_PIXEL_HEIGHT)
      iTy2a = -(rand() % 4)-2;


    if(fncPoint(iMx,iMy) != 0) {/*-----当り判定(若干甘い)*/
      for(i=0 ; i<100 ; i++)
        subLine(iMx  ,iMy  ,rand()  %  224  ,rand()  %  144
,(rand() % 3)+1);

      sprintf(sBuf,"score  =  %lu"  ,sys_get_tick_count()  -
ulTstart);
      text_put_string(5,10,sBuf);  /*ゲームオーバー*/
      sys_wait(75);
      break;
    }

    if((rand() % 50) == 0) {  /*---難易度UP*/
      iTmax++;
      if(iTmax >= (TAMAMAX-1))
        iTmax = (TAMAMAX-1);  /*敵の尻尾を長くする*/
    }

    if( ulTback == sys_get_tick_count() )
      sys_wait(1);  /*ループ速度が早すぎの場合、時間調整*/

    ulTback=sys_get_tick_count();
  }
}
```



PlayStation 2でゲームが作れる！

「BASIC STUDIO パワフルゲーム工房」入門

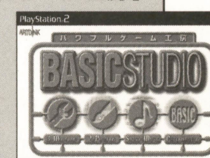
第6回 3Dポリゴンにチャレンジ

文／山崎　寛

先日（7月30日，31日）行われた公開講座の様子をちょっと見学したのですが，中高生を中心としたプログラミングは初めてという方が楽しみながらBASICを扱っていたのが印象的でした。

プログラムは未経験という方も，やる気さえあればきっと大丈夫。BASIC STUDIOでプログラミングに挑戦してみませんか？

『BASIC STUDIO パワフルゲーム工房』
発　売：アートディンク
対応機種：PlayStation 2
メディア：CD-ROM 2枚組
　　　　　発売中
価　格：9,800円（税抜）
http://www.artdink.co.jp/japanese/title/bs/index.html



《第1表》スプライトと3Dモデルの命令対応

命令の内容	スプライト	3Dモデル
表示する	SPRDISP	MODDISP
消去する	SPROFF	MODOFF
内容を指定	SPRTEX	MODSET
位置を指定	SPRPOS	MODPOS

3Dポリゴンを使う

今回の題材は，待ちかねたという方もいるかもしれませんが，3Dのポリゴンモデルを扱ってみたいと思います。

PS2自体が非常に高度な3D機能を持っていることもあり，BASIC STUDIOの3D機能には注目している方も多いことでしょう。

実際のところ，BASIC STUDIOでは，非常に簡単に3Dポリゴンを扱うことができます。実際に3Dモデルを表示して試してみましょう。

```
MODDISP 1
```

と実行してみてください。**写真1**のようにポリゴンの3Dモデルが表示されましたね。

BASIC STUDIOでは，3Dモデルもスプライトと同様，それぞれに番号をつけて管理します。今実行したのは，「1番の3Dモデルを表示しなさい」ということになります。

3Dモデルとスプライトに関しては，ほぼ同様の命令が揃っています。命令の対応については**第1表**を見てください。

つぎに，表示している3Dモデルの内容を変えてみましょう。試しに

```
MODSET 1,6,1
```

と実行すると，**写真2**のように剣士が表示されます。このように，スプライト同様BASIC STUDIOにはデフォルトでいくつかの3Dモデルが入っていて，すぐにゲームで使うことができます。もちろん，POLY-CRAFTで作成したデータを読み込むことも可能です。

あらかじめ入っている3Dモデルデータは，**第1図**のようになっています。

今月のサンプル・ゲーム

早速3Dモデルを使った簡単なゲームを作ってみました。

主人公が四角いマップの上を移動していくので，パッドの上下左右で主人公を動かし，パッドの○ボタンでジャンプします。マップから落ちるとゲーム・オーバーです（**写真3**，**写真4**）。

内容的には普通の2Dゲームと大差ないのですが，ちょっと見た目が変わっただけでもずいぶん印象が変わります。

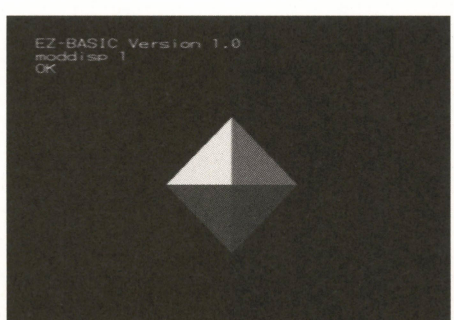
このプログラムでは，270行～280行でカメラの設定をしているのですが，ここを改造することによってゲームの見た目が大きく変わります。試しに，

```
270 CAMPOS 1,0,-600,0
280 CAMROT 1,3.14/2,0,0
```

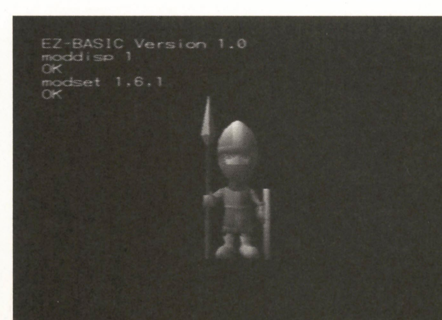
とすると**写真5**のように真上から見下ろした形となり，2Dゲームに非常に近くなります。

逆に，

```
270 CAMPOS 1,0,-100,-350
280 CAMROT 1,3.14/8,0,0
```

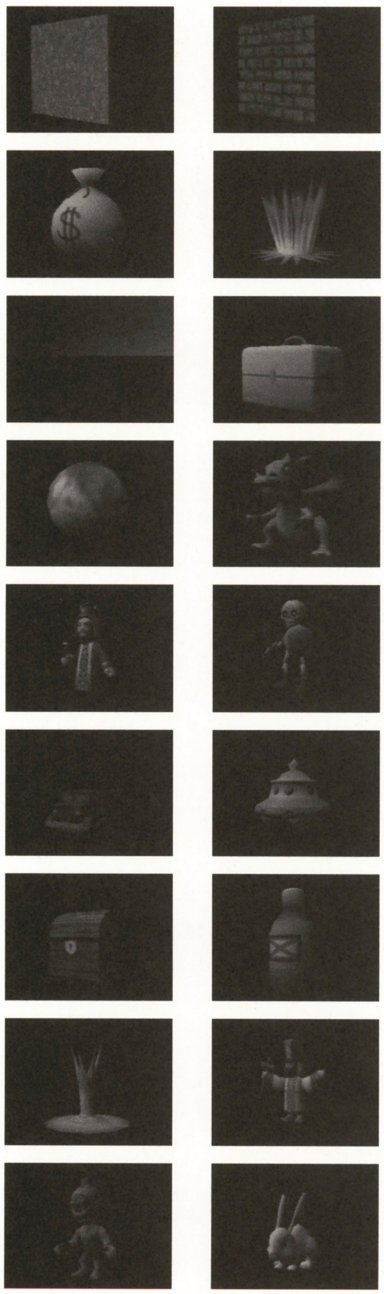


《写真1》画面に3Dモデルが表示された

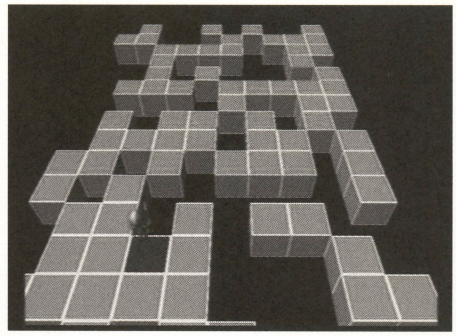


《写真2》3Dモデルの種類を変えてみる

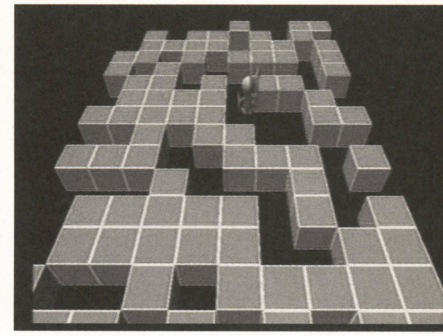
デフォルトの3Dモデルたち



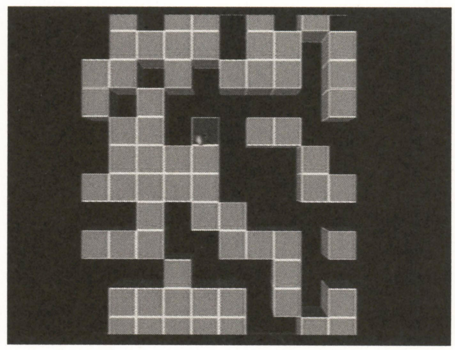
《第1図》このような3Dモデルがデフォルトで入っているぞ。これは数ある中のほんの一部。ぜひ活用しよう！



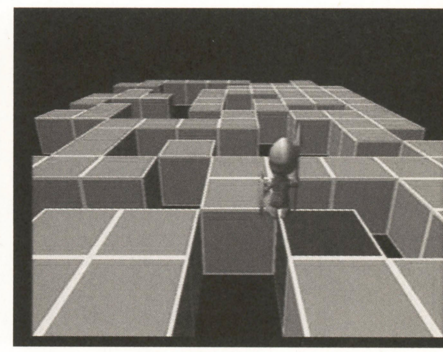
《写真3》サンプルゲームの画面



《写真4》パッドの○ボタンでジャンプ

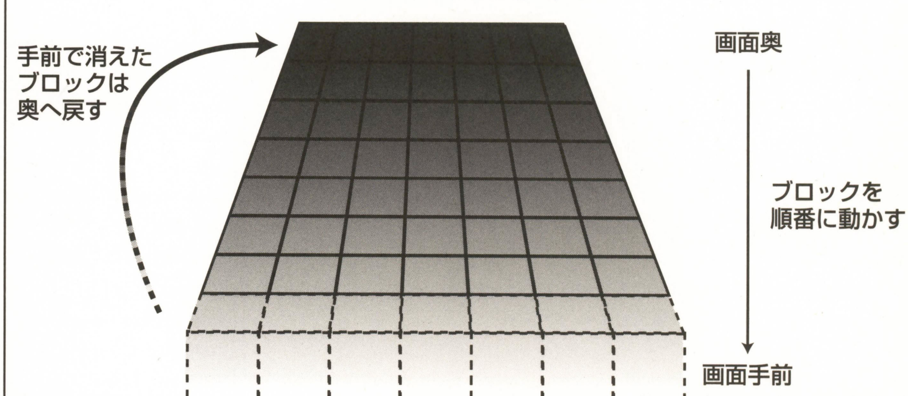


《写真5》真上からの視点で2Dゲームに



《写真6》視点が低いと迫力が増す

スクロールのしくみ



《第2図》ブロックは順番に手前に動く。一番手前までくると消え，消えたブロックは一番奥へと戻る

として視点を下げると，先が見づらくなりますが迫力が増します（写真6）。

プログラムの仕組み

今回のサンプルでは，横10×縦15のブロックを順番に動かすことでスクロールさせています。スクロールして下に消えたブロックは再び上に戻し，ぐるぐるとループさせています（第2図）。

また，それぞれの場所のブロックの有無は，配列変数MAPに入れて判定に利用しています。

150行～160行，190行～200行では，主人公の下にあるブロックを異なる種類にして，画面で色違いで表示されるようにしています。ジャンプしたときにどこにいるのかわかりにくかったため，目安としての表示です。本当は影を表示したかったのですが，簡単に影を表示するいい方法を見つけられなかったため，このようにしてみました。

改造について

また，このサンプルではスコア表示もなく，いくら進んでも難易度が変わりません。そこで，改造を考える方はそのあたりを改良するといいでしょう。

スコアは進んだ距離が変数SLに入っているので，これを使うといいでしょう。難易度は630行のRNDの後にある「＊2」の数値を大きくすることで，四角いブロックの数が減って難しくなりますから，このあたりを変化させてみてはいかがでしょうか。

《リスト1》サンプルゲームのプログラム・リスト

```
100 !●サンプルリスト
110 CALL SYOKI
120 !●メインルーチン
130 DO
140   CALL SCROLL
150   CALL BASYO
160   MODSET 100+X+Z*FX,2,9 ―― 足下のブロックを元に戻す
170   CALL IDOU
180   CALL SC
190   CALL BASYO
200   MODSET 100+X+Z*FX,2,10 ―― 足下のブロックの色を変える
210   CALL HANTEI
220   FSYNC 1
230 LOOP
240 !●初期設定
250 SUB SYOKI
260 INIT
270 CAMPOS 1,0,-300,-380  ┐
280 CAMROT 1,3.14/4,0,0   ┘― カメラの設定
290 MODROT 1,0,3.14,0 ―― 主人公を後ろ向きにする
300 MODSET 1,6,1 ―― 主人公の形をセット
310 MODDISP 1 ―― 主人公表示
320 FX=10 ┐
330 FZ=15 ┘― ブロックの横，たての個数
340 DIM MAP(FX+1,FZ+1)
350 MX=0  ┐
360 MZ=200├― 主人公の位置
370 MY=0  ┘
380 JDY=0 ―― ジャンプ用
390 SL=0 ―― 歩いた距離
400 FOR I=100 TO FX*FZ+99 ┐
410   MODSET I,2,9        │
420   MODDISP I           ├ ブロックの形をセットして表示
430 NEXT I                ┘
440 END SUB
450 !●ブロック表示
460 SUB SCROLL
470 TZ=FZ*20-MOD(SL,FZ*40) ―― 表示をはじめる場所を求める
480 K=100+FZ*FX-1 ―― 表示するブロックの番号
490 FOR J=0 TO FZ-1
500   TX=FX*20-40
510   FOR I=0 TO FX-1    ┐
520     MODPOS K,TX,0,TZ │
530     TX=TX-40         ├ ブロックを順番に横に並べる
540     K=K-1            │
550   NEXT I             ┘
560   TZ=TZ-40
570   IF TZ<-FZ*20 THEN TZ=TZ+FZ*40 ―― 画面手前が終わったら奥に戻す
580 NEXT J
590 IF MOD(SL,40)=39 THEN ―― スクロールして画面から消えた部分の処理
600   TZ=MOD(INT(SL/40),FZ) ―― マップ上での場所
610   K=TZ*FX+100 ―― 3Dモデルの番号
620   FOR I=0 TO FX-1
630     A=INT(RND*2) ―― ブロックの有無を乱数で決定
640     MAP(I+1,TZ+1)=A
650     IF A=0 THEN  ┐
660       MODDISP K  │
670     ELSE         ├ ブロックの有無に従い表示／非表示を切り替え
680       MODOFF K   │
690     END IF       ┘
700     K=K+1
710   NEXT I
720 END IF
730 END SUB
740 !●移動
750 SUB IDOU
760 TX=MX ┐
770 TZ=MZ ┘― 現在位置を保存
780 IF PAD(1,4)>0 THEN MZ=MZ-2 ┐
790 IF PAD(1,2)>0 THEN MX=MX-2 │
800 IF PAD(1,3)>0 THEN MZ=MZ+2 ├ パッドで上下左右に移動
810 IF PAD(1,1)>0 THEN MX=MX+2 ┘
820 IF ABS(MX+21)>FX*20 THEN MX=TX ―― 左右の端で止める
830 IF MY<0 THEN MY=MY+JDY/2                   ┐
840 IF PAD(1,6)>0 AND MY=0 AND BTN=1 THEN      │
850   JDY=-20                                  │
860   MY=MY+JDY/2                              ├ ジャンプの処理
870 END IF                                     │
880 IF PAD(1,6)=0 THEN BTN=1 ELSE BTN=0        │
890 JDY=JDY+1                                  ┘
900 CALL BASYO                                 ┐
910 IF MY=0 AND MAP(X+1,Z+1)>0 THEN            │ 移動先にブロックが
920   MX=TX                                    ├ なかったら元に戻す
930   MZ=TZ                                    │
940 END IF                                     ┘
950 MODPOS 1,MX,MY-40,MOD(MZ-SL,FZ*40)-FZ*20 ―― 主人公の位置設定
960 END SUB
970 !●ブロックから落ちたかどうか判定
980 SUB HANTEI
990 IF MAP(X+1,Z+1)>0 AND MY=0 THEN GOTO *OVER ―― ジャンプ中でなく，ブロックの
                                                 上でなければゲーム・オーバー
1000 END SUB
1010 !●現在どのブロックにいるか
1020 SUB BASYO
1030 X=INT((MX+220)/40)      ┐
1040 Z=MOD(INT((MZ-20)/40),FZ)┘― マップ上の位置を計算
1050 END SUB
1060 !●スクロール処理
1070 SUB SC
1080 SL=SL+1 ┐
1090 MZ=MZ+1 ┘― 1ドット動かす
1100 CALL BASYO
1110 IF MY=0 AND MAP(X+1,Z+1)>0 THEN ┐
1120   MZ=MZ-1                       │ ブロックがない場合，主人公を
1130   IF MZ<SL THEN GOTO *OVER      ├ スクロールでひきずる
1140 END IF                          ┘
1150 END SUB
1160 !●ゲーム・オーバー
1170 LABEL OVER
1180 MZ=MZ+1
1190 FOR Y=-40 TO 0
1200   MODPOS 1,MX,Y,MOD(MZ-SL,FZ*40)-FZ*20
1210   FSYNC 1
1220 NEXT Y
1230 LOCATE 13,11
1240 PRINT "GAME  OVER"
1250 END
```

超 初心者向けJava入門 魁!! Dr.JのJAVA-WALK（ジャバウォック）



第4回 プログラム…っぽくなってきたかな？

ワシはワシである！

やぁ，諸君！Dr.Dの従兄弟こと，Dr.J（ドクトルジェイ）である！なに，いつもいつも同じ書き出しでもう飽きた，だと？なになに，わしの正体がイトコ殿だとか小娘だとか若造だとか諸説紛々，そもそもDr.Jなど存在しないのではないか，と？

…諸君，ワシの考えかたというか，基本姿勢を述べておこうではないか。それはな，「科学する心」「探求心」である！「こうしたらどうなるだろう」「こうすればこうなる，ということは，ああすればああなるのではないか？」「とにかくやってみよう」といった考えかたこそ，プログラミングをするうえでヒッジョーに大事なコトなのである！

そう考えるとどうかな，ワシの正体なんぞワシ以外の何者でもないコトくらいすぐにわかるじゃろ？その証明～『シュレディンガーの猫』って知っておるかな？箱の中にいる猫が生きとるか死んどるかは，ふたを開ける＝観察するまで状態は確定しない，すなわち箱のふたが閉まった状態では，猫チャンは生きておりかつ死んでおる，というものじゃ。



そう，もうわかったかの？読者諸君がベーマガ編集部に来て，ワシ本人に会うまではワシはワシでありかつワシではないのだ…ってなんだかワケがわからんくなっておろーが！

まずは前回の宿題から！

では前回の宿題「1から100までの数を足して，答えを表示する」の，答えからいってみよう！ヒントは「繰り返し文」

```
for (初期値設定;条件判定;条件更新){
    繰り返される処理
}
```

だったワケだが…

ファイル　Syukudai_JW10.java

```
public class Syukudai_JW10 {
    public static void main(String args[]) {
        int counter,result;
        result=0;
        for (counter=1;counter<=10;counter++){
            result=result+counter;
        }
        System.out.println("Kotae = " + result);
    }
}
```

これを実行すると…

```
C:¥javawalk>javac Syukudai_JW10.java

C:¥javawalk>java Syukudai_JW10
Kotae = 55
```

さて，どうじゃったかな？ファイル名（＝クラス名）や，変数の名前は読者諸君によってちがうだろうが，大体同じで，エラーが出ていなければ良いのじゃ。というわけで，今回もポイントになる行をチェックしていこう。まずは，

```
int counter,result;
```

じゃ。これは，

int ……… Integerの略で，整数の型を宣言している。

counter … 変数。数値データの入れ物だ。頭にintを付けたので，「整数の変数counter」という意味になっておる。

result …… これも同じく変数。同じ型の宣言を当てはめるときは，","(カンマ)でつなげてしまえばイチイチ，

```
int counter;
int result;
```

と書かなくてすむのじゃ。何事もラクをしよう！…

じゃが，読みにくくなるので，一度に書くのはせいぜい5個くらいにしておこう。

そのつぎの，

```
result=0;
```

は，result の初期値として0を入れ（代入し）ておるのだ。ちなみに，この行を忘れるとどうなるかというと…，

```
C:¥javawalk>javac Syukudai_JW10.java
Syukudai_JW10.java:5: 変数 result は初期化されていない可能性があります。
                result=result+counter;
                      ^
Syukudai_JW10.java:7: 変数 result は初期化されていない可能性があります。
            System.out.println("Kotae = " + result);
                                            ^
```

となるのだ。そりゃそうだ，

```
result=result+counter;
```

という処理ならば，最初にresultの値が決まっていなければ足し算の結果がどうなるかは不確定だ。一般に，初期値のない場合でもコンパイラは初期値を0にしてくれる場合が多いようだが，賢明なる読者諸君は初期値設定を忘れずに行おうではないか。

ちなみに，初期値設定は型を宣言するときに同時にしてしまっても構わないぞ。

```
int counter,result=0;
```

と書けばカンタンである！

さて，肝心の繰り返し処理だが，

```
for (counter=1;counter<=10;counter++){
```

に対して，

counter=1 …… 初期値設定として，counterに1を代入している

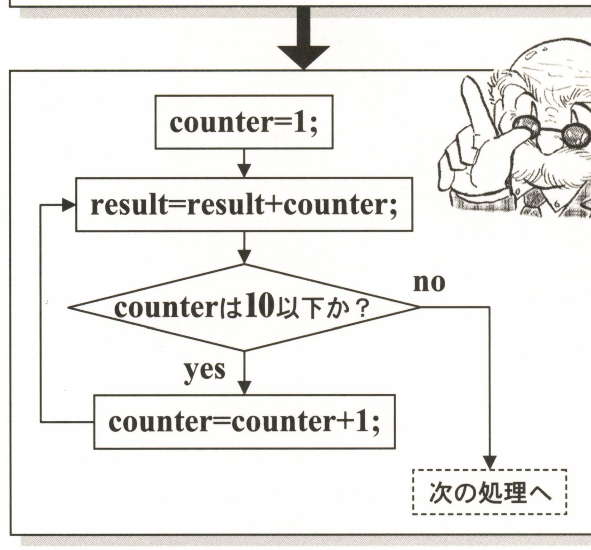
counter<=10 … 終了条件として，counterが10以下である限り処理をする

counter++ …… 条件の更新として，処理が終わると1をたす。これはちょっと特殊な書きかたで，C言語などではおなじみの演算子とよばれるものなのだ。counter=counter+1 と同じ意味。

すなわち，こういう処理なのである。フローチャート風にしてみよう（**第1図**）。

これ自身はあまりJAVAっぽくない処理なのだが，今時のプログラムの基本であり，大事なコトなのである。よく理解しておくのじゃよ。

```
for (counter=1;counter<=10;counter++){
      result=result+counter;
}
```



《第1図》繰り返し処理

そろそろ任意の入力をしてみよう

あああ，なんだか宿題の解説が長くなってしまったのう。今回のメインはそれだけで終わりじゃないのじゃよ～！これまではあらかじめ処理をプログラムの中に全部書いておいて，実行すれば答えが出るというタイプのものばかりじゃった。が，それだけではつまらんよな！そこで，いよいよ今回はプログラムを実行してからデータの入力を行うプログラムを作ってみよう。



ファイル　Toshizan2.java

```
import java.io.*;

public class Toshizan2 {
    public static void main(String args[]) throws
  IOException{
        int year_tsugumi, year_kage, year_doctor;
        String str;

        BufferedReader keybuffer=new BufferedReader
                (new InputStreamReader(System.in));
        System.out.println("Tsugumi no Toshi ha ?");
        str=keybuffer.readLine();
        year_tsugumi=Integer.parseInt(str);
```

```
        System.out.println("Kage no Toshi ha ?");
        str=keybuffer.readLine();
        year_kage=Integer.parseInt(str);

        year_doctor=year_tsugumi+year_kage+10;
        System.out.println("Dr.D no Toshi ha ..." +
  year_doctor);
    }
}
```

およよ，急に難しくなったかな？？？？？じつはたくさんこのプログラムには秘密(?)が隠されているのだが，まずは実行してみよう！

```
C:¥javawalk>javac Toshizan2.java

C:¥javawalk>java Toshizan2
Tsugumi no Toshi ha ?
20 ◀……………… キーボード入力しています
Kage no Toshi ha ?
35 ◀……………… キーボード入力しています
Dr.D no Toshi ha ...65

C:¥javawalk>
```

おお！できておる！できてるのはいいのだが，やっぱりプログラムに書かれておる意味がさっぱりわからん！う〜〜む…とにかく解説してみよう。まずはコレだが…

```
import java.io.*;
```

これはキーボード入力をするために必須な「おまじない」だ！本当は入力用の「クラス・パッケージ」なるものをimportすなわち輸入というか取り込んでいるのだが，まぁ，まずはおまじないとして覚えておこう！

つぎはいつものメイン文だが，これもよけいなものがくっついておる。

```
public static void main(String args[]) {
```

throws IOException　とは，じつは読んで英語のごとし…IO（アイオー，Input/Outputの略，入出力のこと）で起きる例外(Exception)を投げる(throws)ということ，すなわち，入出力（今回の場合はキーボード）で発生する例外（早い話がエラーのこと）をJAVAのシステムが検知するためのものなのだ。こいつも，やはり「おまじない」として扱おう。

```
String str;
```

これはキーボード入力された文字列を受け取るための文字列変数だ。

```
BufferedReader keybuffer=new BufferedReader
        (newInputStreamReader(System.in));
```

これ！これも今のところは「おまじない」として覚えておこう！keybufferという文字列だけは好きなように変えてかまわないぞ。ここで行われているのは，keybufferではじまる入力の受け取り口を用意しているのだ。これも含めて，くわしくはもっとJAVAの細かいこと，いろいろなことを覚えてから，あらためて解説することとしよう。

```
str=keybuffer.readLine();
```

そして，これが実際にキーボードの入力データを代入している部分だ。

```
year_tsugumi=Integer.parseInt(str);
```

最後に，文字列strの中身を，Integer.parseIntを使って整数に置き換えて（型変換）いるのである。

改造するココロはプログラム学習の基本のココロ

それでは次回までのあらたな宿題じゃ！おまじないはおまじないとして，これでキーボード入力をプログラムで使えるわけだ！読者諸君はこれまでのサンプル・プログラムをいろいろ改造してバリエーションを作ってみてくれたまえ！プログラムの上達の基本は，何はともあれ「マネをして」「自分なりに改造して」を繰り返すことにつきる！

今年は台風の当たり年のようじゃが，荒天で外出できない秋の夜長，じっくりプログラミングにハゲんではいかがだろうか！なに，ワシに言われたくないと？余計なお世話じゃの！では，また来月じゃ！

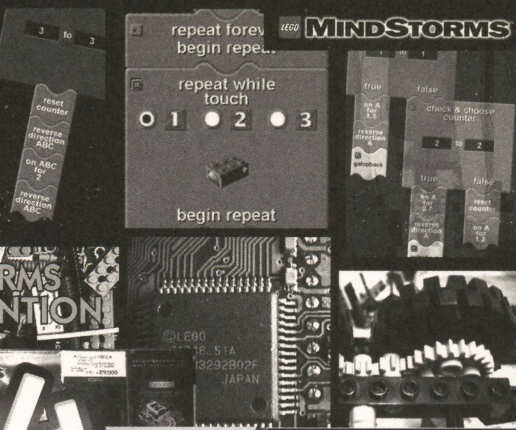


パソコンでロボットを操る！

レゴ道場 桐林部屋

チャレンジ!! レゴ・マインドストーム

LEGO MINDSTORMS ROBOTICS INVENTION SYSTEM



お題：「ライントレース競技会」参加のロボットたち

こんにちは，桐林です。今回は，「ツクモMUGカップ　ライントレース競技会」の報告と，参加したマシンの説明です。

（文／桐林星河）



桐林星河（きりばやしせいが）
機械やロボットが好きな14歳。小学生の頃からメカに慣れ親しみ，LEGO MINDSTORMSの競技などでは常に好成績。TV出演なども多数。現在中学3年の受験生。

盛り上がった「ライントレース競技会」

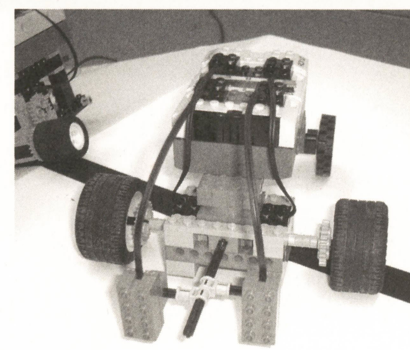
LEGO-MUG（http://www.legomug.gr.jp/）主催による「ツクモMUGカップ　ライントレース競技会」が7月22日，後援のツクモロボコンマガジン館（東京都千代田区外神田3-2-13　☎03-3251-0987）4階特別会場で行われました。

競技会には，ノーマルクラスに8台，エキスパートクラスに6台の参加がありました。そのうち，無事完走したのは，ノーマルクラスが6台，エキスパートクラスが2台という結果でした。結果からもわかるとおり，とても難しいコースで，みなさん苦労されていたようすでした（**第1図**）。くわしい結果は**第1表**を見てください。

今回の見所は，なんといっても「上り下りの坂」と「直角カーブ」でした。どんな方法でクリアするのかとても楽しみに見ていましたが，工夫している人はほとんど見当たらず，そのままラインをトレース（追いかける）しながらなんとかクリアしていきました。ぼくもその1人なのですが…（^^ゞ。

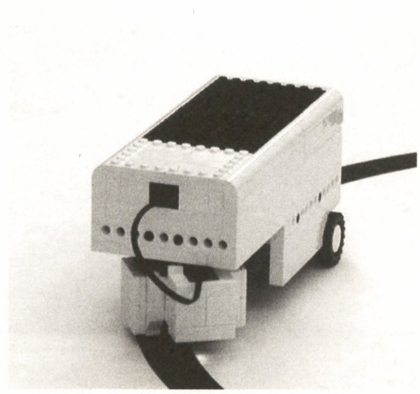
目に付いたロボットたち

ちなみにエキスパートクラス1位のVTRさんの「ぐっと　チャーG」（**写真1**）は，直角カーブだけゆっくり走って，上り坂からは高速になるようなプログラムが組まれていて，とても機敏な動きで，「速い!速い!」と参加者を驚かせていました。構造もシンプルで，見た目からも，とても速そうな感じがしました。

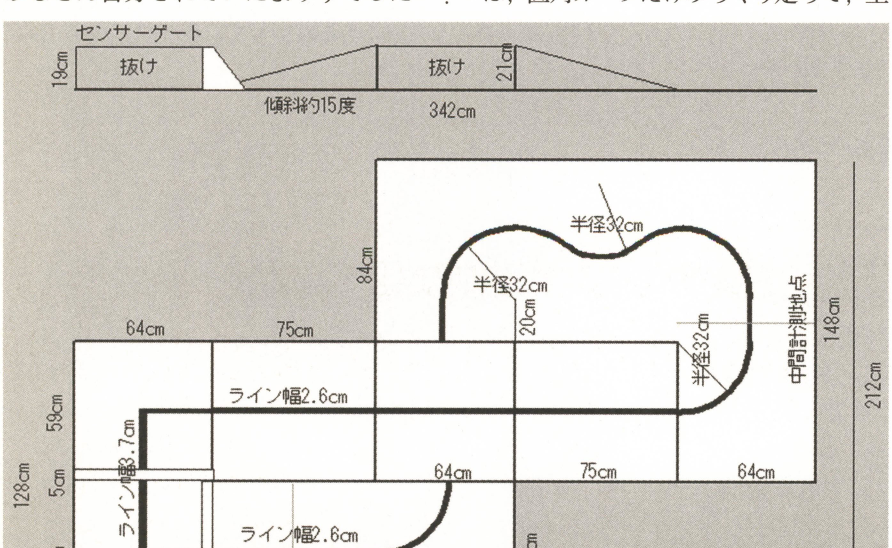


《写真1》VTRさんのマシンです。つくりがシンプルでかっこいいです

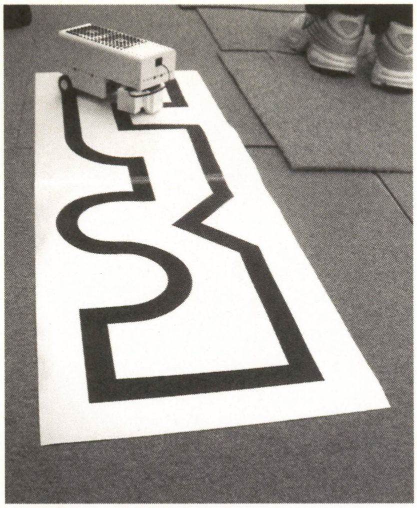
今回の競技で一番目を引いたのは，五十川さんの「じごくのキャンピングトラックこまわりくん」というマシンでした（**写真2**）。なんと，こんな小さなコースを走れてしまうのです（**写真3**）。なぜこんなに小さいコースを走れるかという



《写真2》毎回造形がきれいな五十川さんのマシンです



《第1図》「ツクモMUGカップ」のコースです



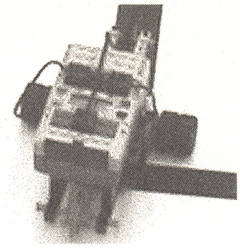
《写真3》五十川さんのマシンはこんなコースを走ってしまいます（前輪駆動だから小回りがきく）

と，前輪でハンドルを切り，なおかつ前輪が駆動しているのです。だから，後輪がバックするような複雑な動きもできるのです。五十川さんは「この小さなコースを走らせるのが目標だった!」のだそうです。

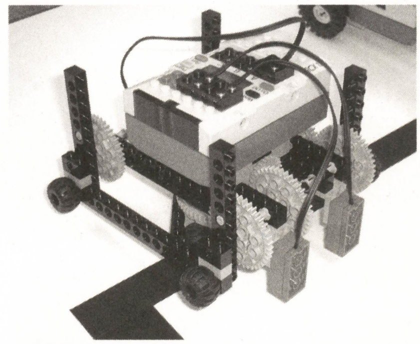
スピードは遅かったのですが，「みんなを驚かせた」という点ではだんぜん1位だと思います。ボディも五十川さんらしい作りになっています。

今回初参加のbupaさんは，プログラムにとても凝っていて，「黒い線の上から外れたときは，マシンをその場で回転させて，また黒い線を見つけて直進」という動作を繰り返していくマシンでした(**写真4**)。この線を見つける動きがまた，おもしろかったのですが，残念ながら完走はできませんでした。でも結構速かったですよ。

また，当日部品を持ってきて，その場



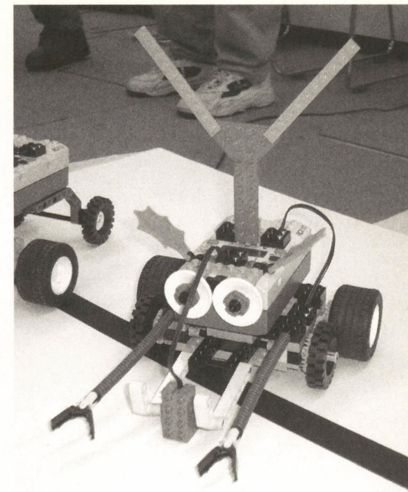
《写真4》bupaさんのマシン。画像が悪いのはご了承ください



《写真5》相原さんのマシン。今回唯一の歩行系

で組み立てて挑んだ相原さんがいます。じつはこのマシン，今回唯一の歩行ロボットだったのですが，とうとうタイム・オーバーでマシンが完成しませんでした。残念(**写真5**)!!　このマシンはおなかにターンテーブルを積んでいて，方向転換できるような仕組みになっていました。

もう1人，驚きの人がいます。MINDSTORMS初心者で今回初参加の酒井さんです。じつは，はじめはRISの説明書に載っているローバーボットで，福西さん（福西さんは今回の競技会のまとめ役）自作のプログラムを使って参加したのですが，少しずつ手を加えていき，なんとノーマル部門1位になりました（**写真6**）。同じマシン，同じプログラムの福西さんよりも，はるかに速かったのです。世の



《写真6》酒井さんのマシン。初参加でなんと1位!!

中は不思議なことも起こるものですね。

酒井さんのように，MINDSTORMS初心者でも楽しめ，しかも優勝できる可能性もあるのですから，MINDSTORMSの奥の深さがわかります。ぜひぜひみなさんもいろんなコンテストに参加してみましょう!!

《第1表》「ツクモMUGカップ ライントレース競技会」完走車の結果

作者	タイム	マシン名
エキスパートクラス		
V T R	16秒17	ぐっと チャーG
桐　林	16秒86	なし
ノーマルクラス		
酒　井	29秒65	W酒井号
桐　林	35秒88	なし
松　原	39秒60	デスレース2001
宇　田	49秒48	なし
福　西	1分11秒36	基本Ⅰ号
五十川	1分39秒98	じごくのキャンピングトラックこまわりくん

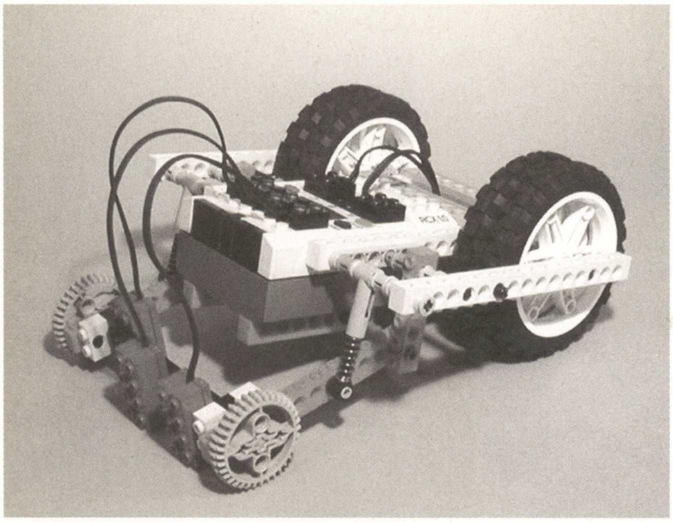
これが競技会参加の桐林製ライントレーサだ！

今回の競技会は同じコースで，RIS付属のプログラムと，モータ，センサのみ（ギヤやビームなどは足してもよい）のノーマルクラスと，プログラム言語は自由，マシンを構成する部品もレゴ社から販売しているものなら何でも使ってよいエキスパートクラスの2部門が用意されました。

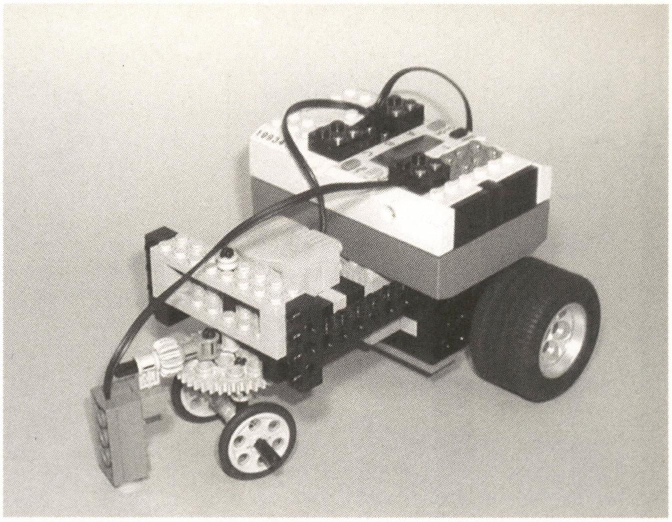
そこで僕の場合は，ノーマルクラスは前号で紹介したステアリング型のマシンに少し手を加えて，エキスパートクラスはマシンを新しく作って挑みました。

エキスパートクラス…プログラムに注目の駆動分散型

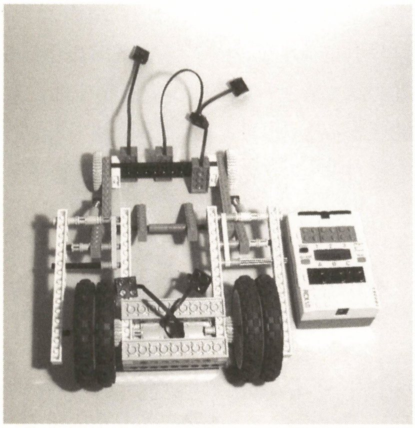
今回は，他のマシンにも部品を使っていたので，モータが2個しか使えない厳しい条件で作らないといけなかったのです。コースには，15度の上り坂があったので，あんまりパワーがないと坂が登れなくなってしまいます。前号で紹介したステアリング式のマシンだと，ステアリングに一つ，駆動に一つモータを使ってしまいます。つまりパワーはモータ1個



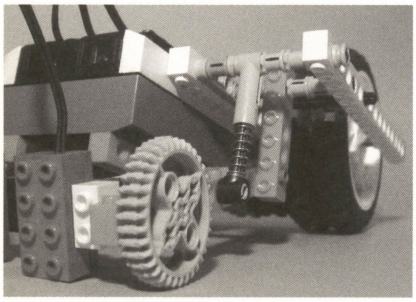
《写真7》エキスパートクラスのマシンです



《写真10》ノーマルクラスのマシン。外見は前号で紹介したマシンと同じ



《写真8》RCXをはずしたところ

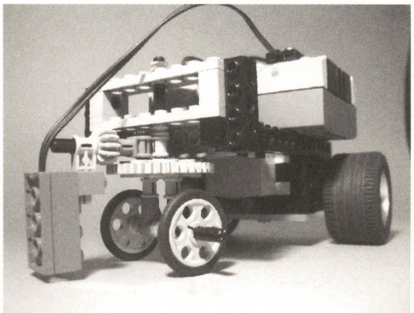


《写真9》前からのアップ。サスペンションが見えます

分になるわけです。

しかし，駆動分散型だと，片輪に1個のモータなので，両輪で2個のモータがつくことになり，パワーが倍になります。そこで今回のマシンは駆動分散型で作りました（**写真7**）。また，RCXも使いまわしができるように，簡単にはずせるようにしました（**写真8**）。

タイヤはレゴの中で一番大きいものを使って，スピードをあげました。でも最終的には，このタイヤが重かったせいで下り坂でのスピードが上がりすぎて，タイムロスになってしまいました。



《写真11》タイヤが幅広になっている点に注目!!

それと同時に出てきた問題が下り坂でのショックです。スピードが上がるせいで，降りた瞬間の衝撃が大きくなってしまい，マシンが何度も壊れてしまいました。そこで，サスペンションをつけてショックを吸収するようにしました（**写真9**）。これでまたマシンが大きくなってしまいました。このままだと，どんどん大きくなってしまいます（^^ゞ）。本当は小さくて軽いほうがいいのですが…。

もう一つ，このマシンはプログラムにこだわってみました。今まで書いてきたように，とてもオーバーラン（ラインからはみ出る）しやすいマシンなので，ラインから外れても走れるようなプログラムを作りました。どうせエキスパートクラスなのだからということで，MINDSTORMSのプログラムの中で，現段階で処理速度が最速のOS「legOS」を使いました。

プログラムは単純で，「左のセンサが黒かったら左へ」「真ん中のセンサが黒かったら直進」「右のセンサが黒かったら右へ」となっています。このプログラムのミソは「センサがすべて白いときは，最後の命令を続行する」というものです。つまり，左が黒くなったのにマシンが曲がり切れずにラインを越してしまい，すべて白いゾーンに入ってしまっても，「左のセンサが黒くなったから，左へ」という命令が続いているので，曲がり続けるというわけです。これが今回のマシンの最大の特徴でした。本番では会場のみんなが驚いてくれました（^_^）。

ノーマルクラス…ステアリング型マシンで挑戦

このマシンのベースは前号で紹介したステアリング型のマシンです（**写真10**）。坂道を乗り越えるためにギヤ比を高く設定して，低速でパワーを上げました。プログラムはRIS2.0で作りました。

このマシンも下り坂で相当スピードが上がってしまい，よくコースアウトしてしまいました。そこで当日，しかたなく前輪の幅を広げてステアリングのスピードを遅く（幅を広げると抵抗が大きくなるため）してみたところ，大成功!! 幅を広げたことで安定性もまして，かなりいい走りができました（**写真11**）。

マシンの分解もなく，無事に走ることができました。

次回は

次回は8月5日に行われた第5回MINDSTORMSフェスティバルに参加したマシンのレポートです。たくさんの作品が出展されていました。どうぞお楽しみに!!

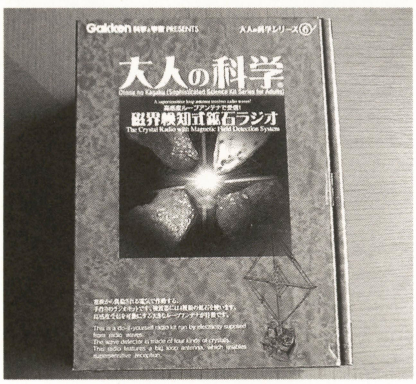
松原拓也の自作一代記

連載7回目 鉱石でラジオを聞こう!

「磁界検知式鉱石ラジオ」の製作



学研の「大人の科学」シリーズは,「エジソン式コップ蓄音機」や「マルコーニ式電波カー」など,往年の「科学と学習シリーズ」の付録を彷彿させる工作キットです。そして,その「大人の科学」シリーズの最新作がこの「磁界検知式鉱石ラジオ」です。内容はコップやニクロム線を使った鉱石ラジオで,工作を楽しみながら検波の原理を学べるというすばらしいキットです。この「大人の科学」という変わった名前は「大人の知的好奇心を満たす手作りの科学キット」という主旨からつけられたものですが,適度な難易度で大人に限らず,あらゆる年齢層にオススメです。（松原拓也）



▲《写真1》学研　大人の科学シリーズ⑥「磁界検知式鉱石ラジオ」定価5,500円（税別）
「大人の科学シリーズ」WEBサイト
http://kids.gakken.co.jp/kit/otona/index.html

組み立て

今回，キットは東急ハンズで購入しました（写真1）が，大きな書店にも置かれているようです。

早速，箱を開けて中身を広げてみました。木のフレームや鉱石のほかに，トランジスタなど電気部品も入っていて，思ったよりもパーツが多いです（写真2）。



▲《写真2》キットの中身。大人向けだけあって，かなりパーツが多い（この他に説明書が付属します）

では，説明書を読みながら組み立てに入ります。道具は最低限，ハサミがあれば作ることができます。パーツの接合はほとんどはめ込むだけ，と簡単です。

各パーツの概要

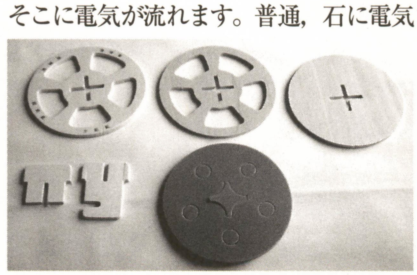
鉱石ラジオは，以下のパーツで構成されています。

●ターンテーブル

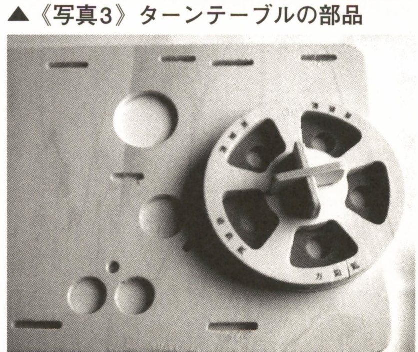
「ターンテーブル」は鉱石を載せるための場所です。回転させて鉱石の位置を変えられます（写真3，4）。

●ピックアップ

「ピックアップ」は，虫ピンと金属板で構成されています。針は鉱石に当てられ，そこに電気が流れます。普通，石に電気は流れないものですが，金属を含んだ鉱石は電気を流すことができます（写真5，6）。



▲《写真3》ターンテーブルの部品



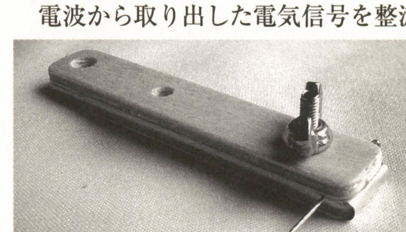
▲《写真4》ターンテーブルの取りつけが完成。スポンジ状の部分に鉱石が載る

●バリコン

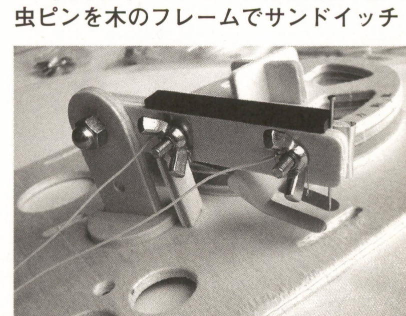
プラスチック製のコップ（2個）の内側と外側にアルミ箔を張って重ねるとバリコン（可変コンデンサ）になります。アルミ箔はビニール張りされていて，電気を流しませんが，その間に電気が溜まって，コンデンサとして機能します。そして，コップの外側を回転させるとアルミ箔の面積が変わって電気容量が変化します。即席の可変コンデンサです。このコップの回転で受信できる周波数が変わり，ラジオの選局ができます（写真7，8，9）。

●配線板

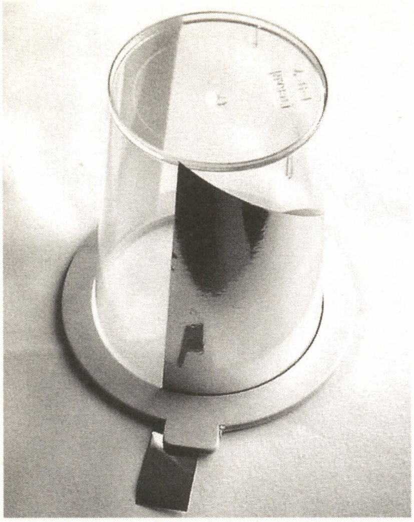
電波から取り出した電気信号を整流



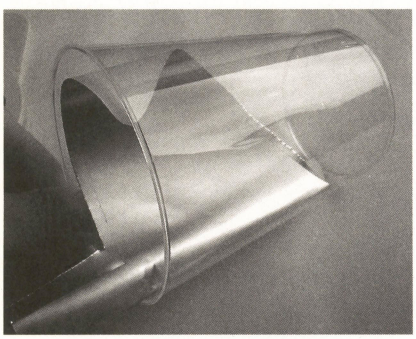
▲《写真5》ピックアップの組み立て。虫ピンを木のフレームでサンドイッチ



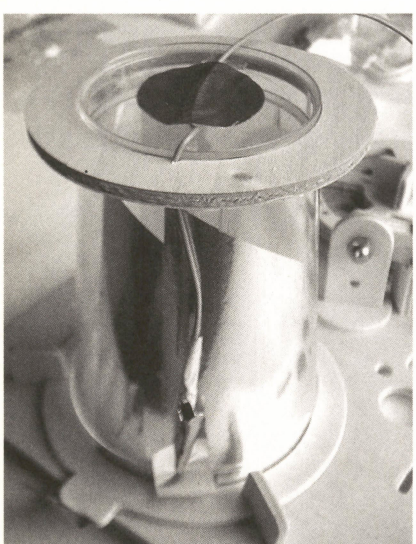
▲《写真6》ピックアップはしっかり止めずにグラグラにして，重みで倒れるようにする



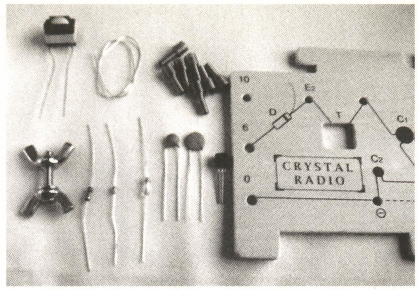
▲《写真7》内側のコップは外向きにアルミ箔を張る



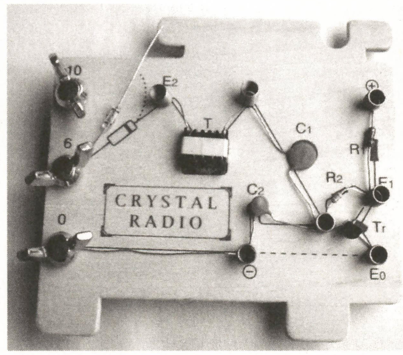
▲《写真8》外側のコップは内側にアルミ箔を張る



▲《写真9》2個のコップを合体させて，バリコンの完成。外側を回転させると，コンデンサの電気容量が変化する



▲《写真10》配線板用の部品。トランジスタ，コンデンサ，トランス，ダイオード，スプリングピンなどで構成されている



▲《写真11》配線盤の完成。はんだ付け不要で組み立てできるのは便利

（プラスの波形だけ抽出）してから，増幅して音波に変換する回路を作ります（**第1図**）。整流に使うダイオードは，一方通行に電気を流す半導体部品です。そして，トランジスタは電流（音量）を増幅する半導体部品です。今回使うトランジスタはNPN型といい，ベース→エミッタ間の電流を数倍に増幅して，コレクタ→エミッタ間の電流に反映します。

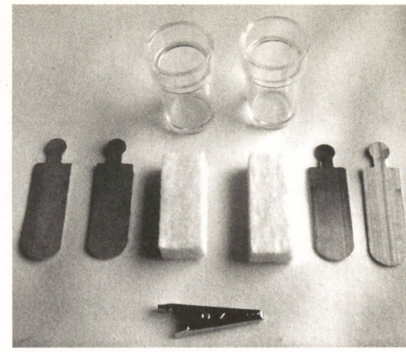
部品の取りつけは，スプリングを押し込んでとめるだけなので，はんだ付けは不要です（**写真10，11**）。

●補助電源部

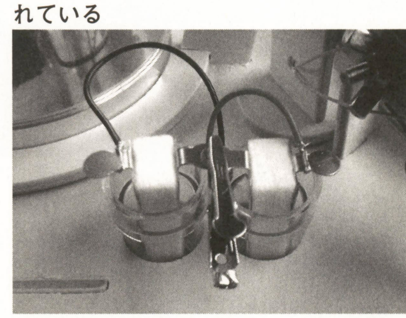
「ボルタ型電池」というタイプです。アルミ板と銅板をフエルトで挟み，試験管に入れたものを2個用意します。そして試験管に食塩水を入れます（食塩水の濃さは200ccの水に対して塩が大さじ1杯くらい）。食塩水によって科学反応が起こり，電気が流れます。なお，塩はキットに付属してませんので，各自の台所から調達してください（**写真12，13**）。



《第1図》波形が音に変換されるまでのイメージ



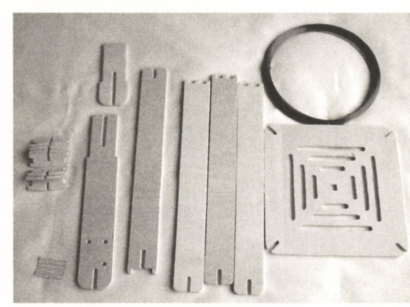
▲《写真12》補助電源用の部品。アルミ板，銅板，試験管，フエルトで構成されている



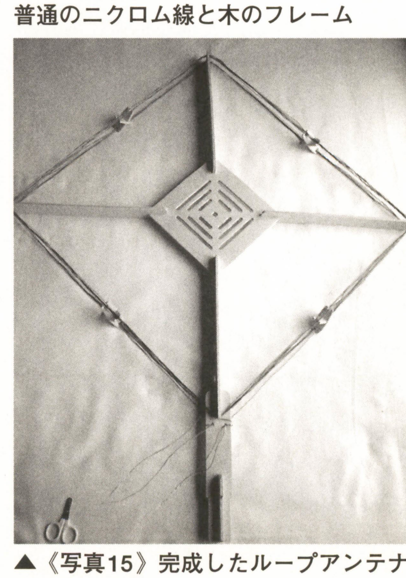
▲《写真13》補助電源の完成。試験管に食塩水を入れると電気が発生する！

●本体

ラジオの土台となる部分です。木の板を組み合わせるだけです。付属の木工用



▲《写真14》ループアンテナの部品。普通のニクロム線と木のフレーム



▲《写真15》完成したループアンテナ。かなり大きい

ボンドで接着することができます。

●アンテナ

「ループアンテナ」というタイプで，十字の骨組みに銅線にエナメル塗料をぬったもの（エナメル線）を巻いただけの簡素なものです（**写真14，15**）。エナメル線を巻く方向は反時計回りで，手前に6周，奥に4周とわけて巻きます。このエナメル線に電波が当たると，その電波の振動する力で，中に電流が発生します。発生する電気は交流で，高速に電位がプラスとマイナスに反転します。その周波数は電波の周波数と同じになります（TBSラジオが954kHz，文化放送が1134kHz）。

●イヤホン

イヤホンはクリスタルイヤホンというタイプです。このイヤホンは高周波をカットしてしまう特性があり，電波の周波数を音の周波数に丸めて鳴らすことができます。イヤホンの接続は，補助電源を使う方法と使わない方法の2種類があります。

……これらの部品を一つに合体させて完成です（**写真16**）。

聞いてみる

このラジオには3とおりの聞きかたがあります。

●ステップ1

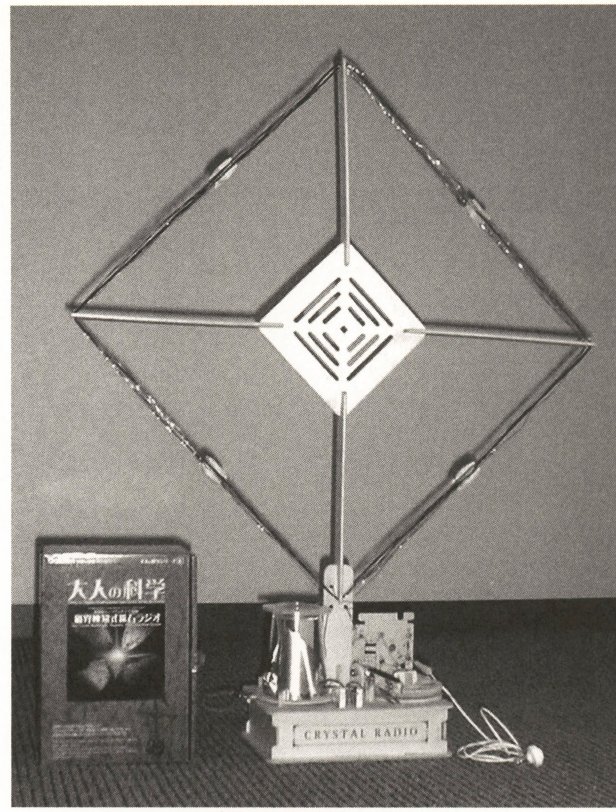
まず，補助電源を使う方法です。試験管に食塩水を入れて電気を起こします。ボルタ型電池の科学反応が進むと，ノイズが聞こえてきます。

つぎにイヤホンを聞きながら，コップ製のバリコンを回転させて選局します。音が聞こえたら，アンテナの向きを変えて，音が大きくなるように調整します。

実際に試してみたのですが，1時間以上，調整を繰り返しても一向に聞こえません。ここでかなり焦ったのですが「もしかして電波が届いていないのでは？」と思い，できるだけラジオを窓辺に置いてみたら，やっとラジオから音楽が聞こえてきました（建物の中では聞こえにくいようです）。何はともあれ成功です。音が小さかったですが，初めてラジオが鳴り出したときは感動しました（**写真17**）。



▲《写真17》窓辺で検波に勤しむ松原。かなり怪しい



▲《写真16》完成した「磁界検知式鉱石ラジオ」！

●ステップ2

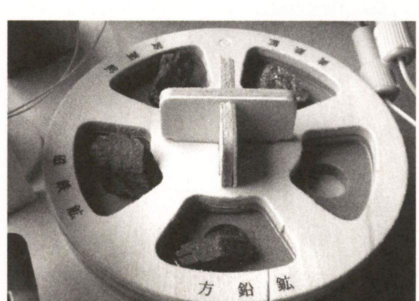
補助電源と鉱石を使う方法です。ここでは，ステップ1で使っていたダイオードを外して，鉱石に置きかえます。まず，ターンテーブルに鉱石を載せます。付属の鉱石は「方鉛鉱」「黄銅鉱」「黄鉄鉱」「磁鉄鉱」の4種類です（**写真18**）。

鉱石の外見は普通の小石のようですが，半導体の特性を持った材質が含まれています。ピックアップの針と金属板を鉱石に当てます。この鉱石と針がダイオードの代わりとなって電流が一方向にのみ流れます。

ターンテーブルに載せるものは鉱石以外にも，表面が少し錆びている10円玉やボルトが使えるようです。

●ステップ3

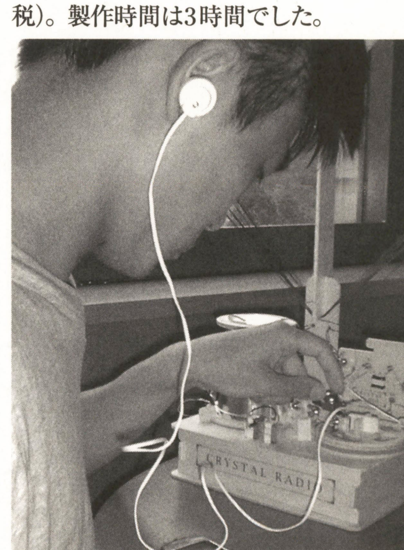
このラジオはゲルマニウム・ラジオのように，電源なしで音を鳴らすことができます。方法は簡単で，増幅回路を通さずイヤホンに直結させるだけです。しかし，増幅せずに電気をそのまま音に変換しているため，かなり音量が小さくなってしまいます。うまくいくかは，電波の受信感度がカギをにぎります（**写真19**）。



▲《写真18》ターンテーブルに鉱石を載せて，鉱石で検波もできる。「方鉛鉱」が一番検波しやすい

まとめ

ぜひ「自作のラジオから音が聞こえてくる」という興奮と感動を味わってみてください。今後も「大人の科学」には注目です。総製作費5,500円（プラス消費税）。製作時間は3時間でした。



▲《写真19》鉱石にピックアップを当てて検波中。調整はかなりシビア

テクノ・トイズ・ラボラトリ
Techno Toys Laboratory

デジタルとアナログの境界線上で作る電子オモチャ

今月の製作

デューパールーパ 完成編

テクノ・オモチャ研究所

TEXT&製作●クワクボリョウタ



デバイス・アーチスト。「ビット」をキーワードにメディアアート作品や電子オモチャなどを製作・発表。ロボコン・ブームのお陰で最近は面白いセンサが店頭に並び，入手しやすくなった。良い傾向です。
www.vector-scan.com

研究所からはチャカポコチャカポコとけたたましいリズムが流れ出ている。そう，それは来なかった近未来からやってきた謎のドラムマシン，「デューパールーパ」のうぶ声なのだ！…というわけで前回からの続き。デューパールーパの完成編です。

今回の改良点

前回，試作編として基本的な動作は確認できたので，今回は最終的な機能として以下の2点を追加したいと思う。

1.テンポやステップ数を変更できるようにする

2.スリープとウェイクアップの追加

いずれも前回製作したものに多少手を加えるだけでOKだ。

テンポ，ステップ数設定スイッチの追加

試作編では140BPMに固定だったテンポを変更できるようにし，また，ステップ数も16に固定だったのでこれも変更できるようにしよう。

そのためにロータリスイッチを買ってきた。日本開閉器のDR-SR16という型番だ。ロータリスイッチは**写真1**のような外観で，ツマミに16進数で0～Fが書いてある。端子はコモン（共通）が一つと，1,2,4,8と書かれた端子の計5本が出ている。だいたい察しがつくだろうけど，このスイッチはツマミを合わせた数字の値を示すように四つの端子とコモン端子のON/OFF（導通/非導通）が切り替わる。動作の状況は**第1図**のとおり。

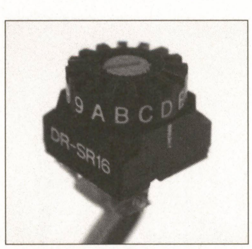
ロータリスイッチは0h～fhまでの値を指定できるので，16とおりの設定が可能だ。選択可能なステップ数は8,10,12,16の4とおりとし，それぞれのステップ数でテンポを80,100,120,140BPMに切り替えられるようにした。まとめると**第1表**のようになる。

回路の追加

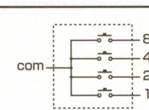
前回作った回路ではPICのI/Oポートが随分余っているので，RB0～RB3を使ってスイッチの値を読ませることにしよう。RB0にスイッチの"1"と書かれた端子をつなぎ，RB1に"2"端子，RB2に"4"端子，RB3に"8"端子をつなぐ。PORTBはプログラムによって内部プルアップができるからそれを使い，コモン端子はグラウンドにつなぐ。**第2図**が回路図，**第3図**が実体配線図だ。

このスイッチは開閉によって2進数を表すのであって，電圧のHi/Loを直接出すわけではないから，入力ポートに抵抗を介してあらかじめ電圧をかけておいて，スイッチの開閉によってそれぞれのピンの電圧を落とすようにしている。内部プルアップ機能はこのような用途のためにPICに用意された機能だ。OPTION_REGレジスタの$\overline{\mathrm{RBPU}}$というビットを0にするとPORTBの入力に設定されたピンすべてにプルアップがかかる（**第4図**）。これにより，何も接続しないときには電気的にHiになる。

ところで，PORTBで読むデータは，2進数の"0"がVcc（電源電圧），"1"が0Vとなり，2進数のイメージが反転したものになっている。いわゆる負論理というやつだ。だから必要に応じて読んだ値をプログラム的に反転して使う。PICではcomfという命令でレジスタの反転値が得られ



《写真1》端子が5本あり，16進数のダイアルで値を指定できる



ロータリスイッチの動作

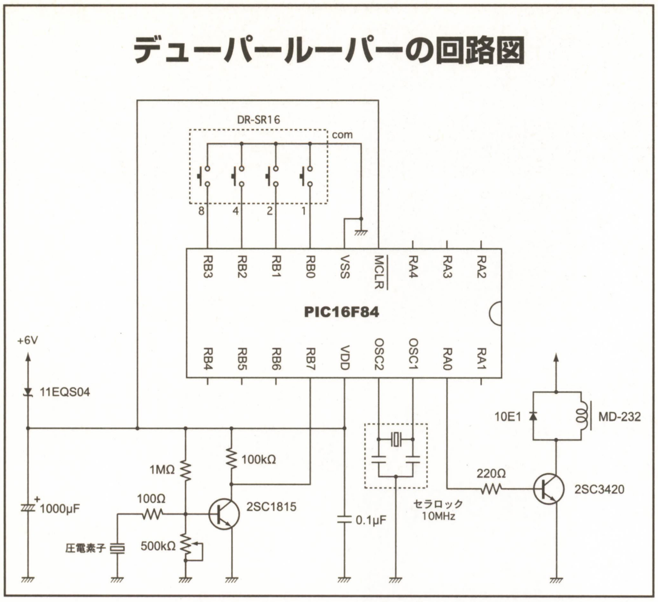
-　：スイッチOFF　●：スイッチON

端子＼ダイアル	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
8	-	-	-	-	-	-	-	-	●	●	●	●	●	●	●	●
4	-	-	-	-	●	●	●	●	-	-	-	-	●	●	●	●
2	-	-	●	●	-	-	●	●	-	-	●	●	-	-	●	●
1	-	●	-	●	-	-●	-	●	-	●	-	●	-	●	-	●

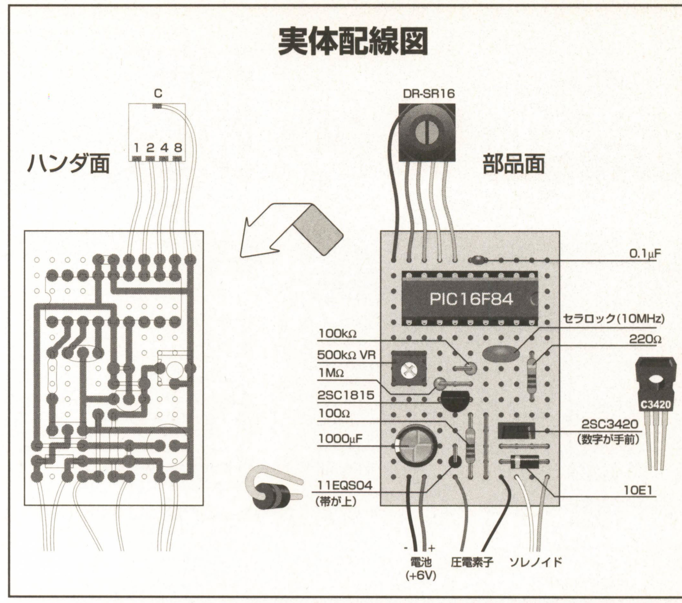
《第1図》DR-SR16の動作状況を示したもの

《第1表》ダイアルの設定

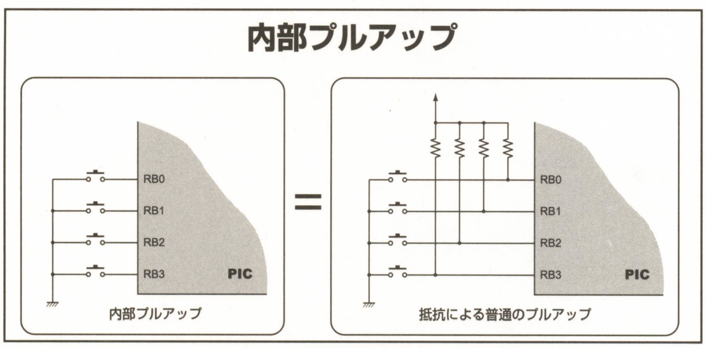
ダイアル	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
ステップ数	8				10				12				16			
BPM	80	100	120	140	80	100	120	140	80	100	120	140	80	100	120	140



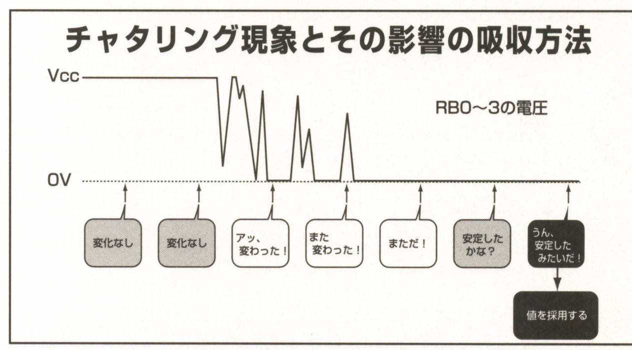
《第2図》図中の↑はともに電池に接続する



《第3図》追加部分はロータリスイッチ。端子付近の刻印に注意して配線しよう



《第4図》OPTION_REGレジスタの$\overline{\text{RBPU}}$の設定で内部プルアップをすると，右図と同様の結果となる。外付け抵抗がなくてすむ



《第5図》RB0〜3の電圧はこんなふうに乱高下するので定期的にスイッチをチェックして，値がいったん変化した後に安定したら，その値を採用する

る。またはffh(2進数で11111111)でxor(排他的論理和)をとればよい。

スイッチの問題点

プログラムでは随時スイッチの状態を読み込み，それに変化があったときに設定を変更する処理を行えばいいのだが，ひとつ厄介な問題がある。それはチャタリングによって不適切な値が飛び込む可能性があることだ。

スイッチはメカなので，ONからOFF，OFFからONに変わる瞬間というのは不安定な状態になる。バタバタとONとOFFを行き来してしまう。これがチャタリングと呼ばれる現象だ。しかも今回使うデジタルスイッチの場合，四つのスイッチがバラバラにこのような動作をする。

これはスイッチがメカである以上どうしても避けられない問題なので，チャタリング防止回路をつけるか，ソフト的に解決する。今回は部品が少なくて済むのでソフト的に対処することにした。

チャタリングを吸収する

やり方は簡単。要は何回かスイッチを読んで，値が落ち着いたらそれを採用すればいいのだ。第5図を見てほしい。まず一定間隔でスイッチを読みに行く。すでにTRIGというフラグが一定間隔で立つように設定されているので，そのタイミングを利用しよう。そして読んだ値を前回読んだ値と比較する。毎回この処理をするたびに専用のカウンタ(sw_count)をデクリメントする。このカウンタが0になったらスイッチの状態が安定したと見なして，その値を採用する。一方，読みの値が違っていた場合にはこのカウンタをリセットして，採用を延期する。最後に次回の比較のために今読んだスイッチの値を保存しておく。

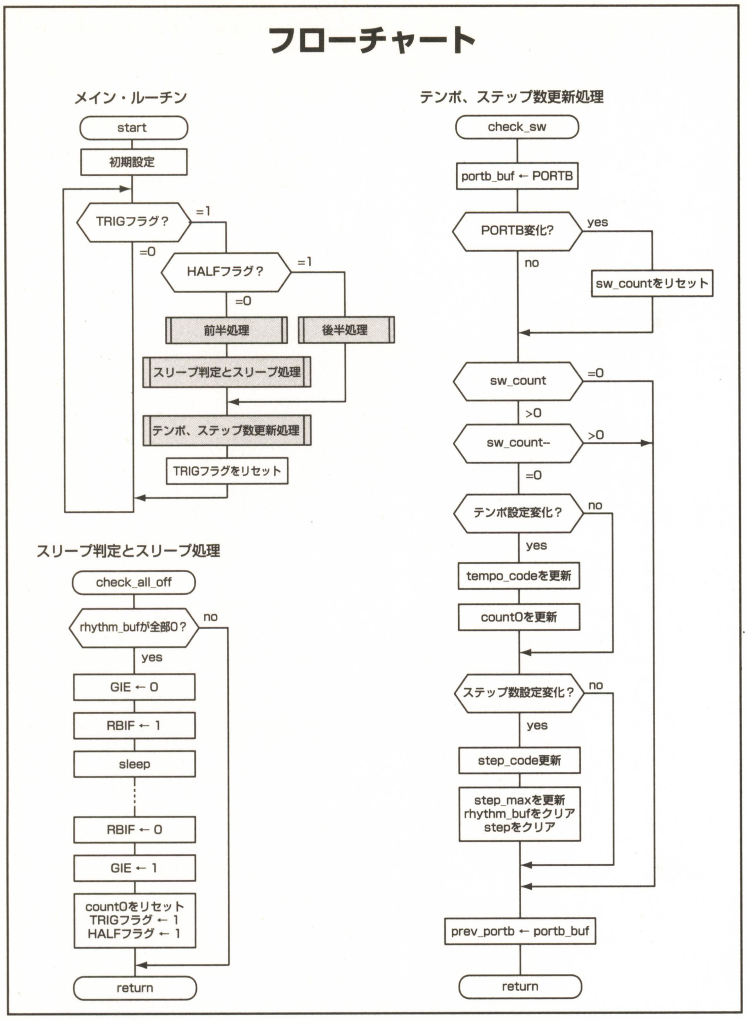
スイッチ入力値の採用が決まったら下位2ビットをテンポ設定として，上位2ビットをステップ数設定としてそれぞれ取り出し，現状と違う値の場合だけ設定変更処理をする。

テンポ変更はcount0_initを書き換える。この変数はTRIGフラグを立てる間隔を決めるものだ。

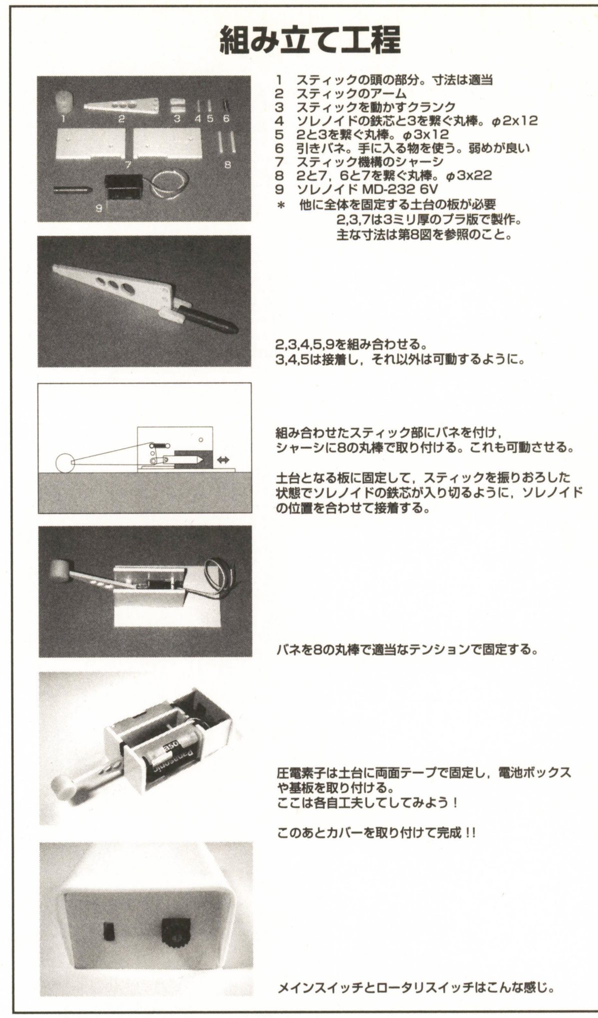
ステップ数変更は総ステップ数を示すstep_maxにあらたなステップ数を入れ，音符情報の入ったrhythm_buf(2バイト)とstep(これは今演奏している位置を表している)を0クリアする。これをしないとこまごまとした不具合が出てしまう。

スリープとウェイク・アップ

入力がなくなってスティックが動作する必要がない状態，つまり完全な沈黙状態ではPICが動く必要がないので，スリープさせてしまおう。スリープ機能とは，



《第6図》変更のあった部分だけをフローチャートにしてみた



《第7図》組み立ては慎重にやろう。自分なりに工夫してみよう

PICがクロックを止めて，ウェイク・アップの条件が満たされるまで休止し，電池の消費を最小限にできる機能だ。

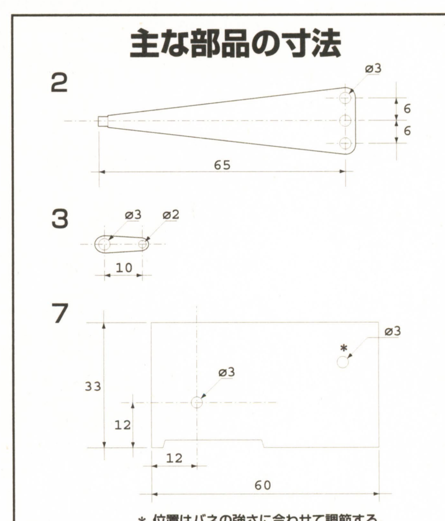
各ステップごとに音符情報(rhythm_buf)をチェックして，すべて休符だったらスリープするようにした。スリープの前には起きる条件を指定しておかないと，永遠の眠りについてしまうので，それもちゃんとやっておこう。ここでは演奏再開＝プレーヤが何かを叩いて衝撃が検出されたときにウェイク・アップするのが望ましいわけで，「RB7〜4のどれかの値が変わったよフラグ」つまりRBIFをウェイク・アップの条件とする。センサはRB7に接続されているので，何か衝撃を検出したらRB7の値が変わり，このRBIFが1になる。今回は割り込み処理に飛ばずに単にウェイク・アップするやり方にした。手順は，まずGIE(全体の割り込み許可フラグ)を0にして全割り込みを禁止する。その後RBIE(RBIFによる割り込み許可フラグ)を1にする。そしてsleep命令を実行してスリープする。全割り込みを禁止しているのにRBIFを立てて意味があるの？　と疑わしく思われる人もいることでしょう。しかし，この設定をすると，RBIEが1になると割り込み処理にジャンプしないけどウェイク・アップしてsleep命令のつぎのアドレスから実行を再開しますよ，という意味になる。

それからウェイク・アップ直後には，count0をリセットし，TRIGとHALFフラグを立てて，すぐに後半処理（前号参照のこと）を始めるように指示している。こうしておくと起き上がった瞬間を基点とした演奏のタイミングが取れるからだ。

以上，変更点をフローにまとめたのが**第6図**。また**リスト1**がそのソースリストだ。書き込み時はHSモード，ウォッチドッグなし，パワーアップタイマあり。

工作の時間

では，スティック機構部分の工作について説明したい。組み立て手順を**第7図**に示そう。これを参考にして作ってほしい。細かい寸法については**第8図**を参照のこと。



《第8図》最低限この寸法を守ればうまく動くはずだ

コツとしては，ソレノイドは鉄心がコイルから離れると飛躍的に吸引力が弱まるので，可動距離はできるだけ短くした

ほうが良いみたい。3mmくらいにとどめると十分な力が得られる。

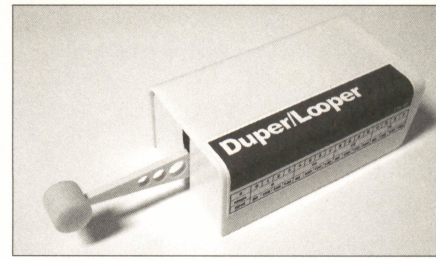
機構ができたら，筐体に収める。自分なりの筐体を作ってみよう！

完成！

写真2が完成したデューパールーパだ！ちなみに，スリープ機能はあるものの，念のため電源スイッチを付けてある。持ち運ぶときに衝撃検出して動作し始めるのも困るし，夜中に突然暴れ出したりしたら怖いからね。

じつは今回のチューンアップは，本機を複数作ったときにその真価をあらわすのだ。たとえば，テンポ設定をデジタル式にしたので，複数のデューパールーパで同期を取らなくてもテンポが合うし，ウェイク・アップした瞬間を演奏タイミングの基点にするので，複数での演奏時に拍のアタマを合わせやすい。さらに8ステップと10ステップの混合など，ポリリズムも演奏することができるという，我ながらマニアックな仕様と相成った。

といったところで，今回は終了！



《写真2》これが完成したデューパールーパだ

《リスト1》完成版デューパールーパのリスト（アセンブラ）

```
;=========================================================================
;
;                       d u p e r   l o o p e r
;
;                 c o d e d   b y   ｸ ﾜ ｸ ﾎ ﾞ ﾘ ｮ ｳ ﾀ
;
;=========================================================================
;
        LIST    C=120, p=16F84, r=dec, x=off, n=0
        include <p16f84.inc>

;======== マクロの定義 ===================================================
push    macro
        movwf   w_temp
        swapf   STATUS,W
        bcf     STATUS,RP0
        movwf   status_temp
        endm

pop     macro
        swapf   status_temp,W
        movwf   STATUS
        swapf   w_temp,F
        swapf   w_temp,W
        endm

;======== シンボルの定義 =================================================
;-------- レジスタ名の定義 --------
temp                    equ     0x0c
w_temp                  equ     0x0d    ; 割り込み時のﾚｼﾞｽﾀ退避用
status_temp             equ     0x0e    ;          //
count0                  equ     0x0f    ; ﾀｲﾐﾝｸﾞ用ｶｳﾝﾀ
count0_init             equ     0x10    ; count0ﾘｾｯﾄ値
tempo_code              equ     0x11    ; ｽｲｯﾁによるﾃﾝﾎﾟ設定値
step                    equ     0x12    ; ﾘｽﾞﾑのｽﾃｯﾌﾟｶｳﾝﾀ
step_max                equ     0x13    ; 総ｽﾃｯﾌﾟ数
step_code               equ     0x14    ; ｽｲｯﾁによるｽﾃｯﾌﾟ数設定値
event                   equ     0x15    ; ｲﾍﾞﾝﾄﾌﾗｸﾞ
stat                    equ     0x16    ; ｽﾃｰﾀｽﾌﾗｸﾞ
portb_buf               equ     0x17    ; PORTBの内容
prev_portb              equ     0x18    ; 前回読んだのPORTBの内容
sw_count                equ     0x19    ; sw内容の確定用ｶｳﾝﾀ
rhythm_buf              equ     0x20    ; ﾘｽﾞﾑ用ﾊﾞｯﾌｧ (2 byte)

;-------- 各変数の初期値 --------
SW_COUNT_INIT           equ     2       ;

;-------- eventのビットマップ --------
TRIG                    equ     0       ;

;-------- statのビットマップ --------
HALF                    equ     0       ;

;-------- PORTAのビットマップ --------
DRIVE_PIN               equ     0       ;

;========  リセットベクタ ===============================================
        org     0x0000                  ; 起動時は0番地から実行を始める
        goto    initialize

;========  割り込みベクタ ===============================================
        org     0x0004                  ; 割り込みが起きると
                                        ;          4番地にジャンプする
;========  割り込みルーチン  ============================================
interrupt:
        push                            ; コンテキストの保存

        decfsz  count0,F                ; count0--==0なら以下を実行
        goto    int_90                  ; それ以外は90へ

        movf    count0_init,W           ; count0の初期化
        movwf   count0                  ;

        bsf     event,TRIG              ; TRIGﾌﾗｸﾞをｾｯﾄ

        movlw   1<<HALF                 ; HALFﾌﾗｸﾞを反転する
        xorwf   stat,F                  ;

int_90:
        bcf     INTCON,T0IF             ; タイマ割り込みフラグをクリア
        pop                             ; コンテキスト復元
        retfie                          ; タイマ割り込み処理終了

;========  初期設定  ====================================================
initialize:
        clrf    PORTA                   ;
        clrf    PORTB                   ;

        bsf     STATUS,RP0              ;

        movlw   b'00000000'             ; PORTAの入出力方向を設定
        movwf   TRISA                   ;

        movlw   b'10001111'             ; PORTBの入出力方向を設定
        movwf   TRISB                   ;

        movlw   b'00010001'             ; TMR0の設定
        movwf   OPTION_REG              ;  (4命令ｻｲｸﾙごとに++)
                                        ; PORTBのﾌﾟﾙｱｯﾌﾟ

        bcf     STATUS,RP0              ;

        movlw   0x50-(temp+1)           ; zero clear 0x50 ~ 0x0d
        movwf   temp                    ;
        movlw   (temp+1)                ;
        movwf   FSR                     ;
clr_lp: clrf    INDF                    ;
        incf    FSR,F                   ;
        decf    temp,F                  ;
        btfss   STATUS,Z                ;
        goto    clr_lp                  ;

        movlw   -1                      ; ﾃﾝﾎﾟ,ｽﾃｯﾌﾟ数の初期化
        movwf   tempo_code              ;
        movwf   step_code               ;
        call    check_sw                ;

        bsf     INTCON,T0IE             ; ﾀｲﾏ割り込みを許可
        bsf     INTCON,GIE              ;
                                        ;
;========  メインルーチン  ==============================================
main:
        btfss   event,TRIG              ; TRIGﾌﾗｸﾞが1なら以下へ
        goto    main                    ; それ以外はmainへ

        btfsc   stat,HALF               ; HALF=0なら前半処理へ
        goto    main_10                 ;
        call    first_half              ;
        call    check_all_off           ;
        goto    main_90                 ;
main_10:                                ;
        call    second_half             ; それ以外は後半処理へ

main_90:
        call    check_sw                ; ﾛｰﾀﾘｽｲｯﾁのﾁｪｯｸ

        bcf     event,TRIG              ; TRIGﾌﾗｸﾞをﾘｾｯﾄ
        goto    main                    ; mainへ戻る

;=========================================================================
first_half:
        bcf     PORTA,DRIVE_PIN         ; ｿﾚﾉｲﾄﾞをOFF

        call    select_memory           ; FSR  <- step>>3
                                        ; temp <- stepの下位3ﾋﾞｯﾄ

        btfss   INTCON,RBIF             ; 入力に変化あり？
        goto    first_half_10           ;       なければ10へ

        movf    temp,W                  ; ﾘｽﾞﾑﾊﾟﾀｰﾝに音符を書き込む
        iorwf   INDF,F                  ;
        goto    first_half_20           ;

first_half_10:
        comf    temp,W                  ; ﾘｽﾞﾑﾊﾟﾀｰﾝに休符を書き込む
        andwf   INDF,F                  ;
        goto    first_half_20           ;

first_half_20:
        movf    PORTB,F                 ; RBIFのﾘｾｯﾄ
        bcf     INTCON,RBIF             ;

        incf    step,F                  ; stepの更新
        movf    step,W                  ; (0〜step_maxを繰り返す)
        subwf   step_max,W              ;
        btfsc   STATUS,Z                ;
        clrf    step                    ;

        return                          ;

;=========================================================================
second_half:
        call    select_memory           ; FSR  <- step>>3
                                        ; temp <- stepの下位3ﾋﾞｯﾄ

        movf    INDF,W                  ; 音符があったら出力をHiに
        andwf   temp,W                  ;
        btfss   STATUS,Z                ;
        bsf     PORTA,DRIVE_PIN         ;

        return                          ;

;=========================================================================
; ｻﾌﾞﾙｰﾁﾝ : select_memory
;         stepの値に応じてｱｸｾｽするﾒﾓﾘのｱﾄﾞﾚｽと、ﾋﾞｯﾄ位置を設定する
;         FSR  <- step >> 3
;         temp <- 3 least bit of step
;=========================================================================
select_memory:
        rrf     step,W                  ; 現在のstepに対応する
        movwf   temp                    ; 音符が格納されているｱﾄﾞﾚｽを算出
        rrf     temp,F                  ; (FSR <- step >> 3)
        rrf     temp,W                  ;
        andlw   b'00011111'             ;
        addlw   rhythm_buf              ;
        movwf   FSR                     ;

        movf    step,W                  ; 現在のstepに対応する
        andlw   b'00000111'             ; 音符が格納されているﾋﾞｯﾄ位置
        call    bit_table               ; (temp <- stepの下位3ビット)
        movwf   temp                    ;

        return

bit_table:                              ; 0〜7の数値を
        addwf   PCL,F                   ;       ﾋﾞｯﾄ位置に変換するﾃｰﾌﾞﾙ
        retlw   1<<7                    ;
        retlw   1<<6                    ;
        retlw   1<<5                    ;
        retlw   1<<4                    ;
        retlw   1<<3                    ;
        retlw   1<<2                    ;
        retlw   1<<1                    ;
        retlw   1<<0                    ;

;=========================================================================
; ｻﾌﾞﾙｰﾁﾝ : check_sw
;         ﾛｰﾀﾘｽｲｯﾁの値（負論理）を読み、以下の条件に応じて処理する
;         ﾃﾝﾎﾟ設定(第1,0ﾋﾞｯﾄ)が変化 ----> tempo_code, count0_initを更新
;         ｽﾃｯﾌﾟ数設定(第3,2ﾋﾞｯﾄ)が変化 -> step_code, step_maxを更新
;                                         rhythm_bufをｸﾘｱ
;                                         stepを0にﾘｾｯﾄ
;=========================================================================
check_sw:
        movf    PORTB,W                 ; PORTBの内容をportb_bufにｺﾋﾟｰし、
        movwf   portb_buf               ;       prev_portbと比較する
        xorwf   prev_portb,W            ;

        movlw   SW_COUNT_INIT           ; 内容が違っていたら
        btfss   STATUS,Z                ;       sw_countをﾘｾｯﾄ
        movwf   sw_count                ;

        movf    sw_count,F              ; sw_countが0なら90へｼﾞｬﾝﾌﾟ
        btfsc   STATUS,Z                ;       それ以外は以下へ
        goto    check_sw_90             ;

        decfsz  sw_count,F              ; sw_count--
        goto    check_sw_90             ;       0になったら以下へ
                                        ;       それ以外は90へｼﾞｬﾝﾌﾟ

        ;-------- check tempo -----------
        comf    portb_buf,W             ; temp <- !(portb_bufの第1,0ﾋﾞｯﾄ)
        andlw   b'00000011'             ;
        movwf   temp                    ;

        subwf   tempo_code,W            ; ﾃﾝﾎﾟ設定に変化があれば以下へ
        btfsc   STATUS,Z                ;
        goto    check_sw_10             ; なければ10へ

        movf    temp,W                  ; tempo_code <- temp
        movwf   tempo_code              ;

        call    tempo_table             ; count0_init <- tempo_table[W]
        movwf   count0_init             ;

check_sw_10:
        ;-------- check step ------------
        rrf     portb_buf,W             ; temp
        movwf   temp                    ;  <- !(portb_bufの第3,2ﾋﾞｯﾄ)>>2
        rrf     temp,W                  ;
        xorlw   b'11111111'             ;
        andlw   b'00000011'             ;
        movwf   temp                    ;

        subwf   step_code,W             ; ｽﾃｯﾌﾟ数設定に変化があれば以下へ
        btfsc   STATUS,Z                ;
        goto    check_sw_90             ; なければ90へ

        movf    temp,W                  ; step_code <- temp
        movwf   step_code               ;

        call    step_table              ; step_max <- step_table[W]
        movwf   step_max                ;

        clrf    rhythm_buf              ; rhythm_bufをｸﾘｱ
        clrf    rhythm_buf+1            ;

        clrf    step                    ; stepを0にﾘｾｯﾄ

check_sw_90:
        movf    portb_buf,W             ; 次回の比較のために
        movwf   prev_portb              ;       prev_portb <- portb_buf

        return

        ;-------- 0〜3に応じてﾃﾝﾎﾟ設定用の数値を返すﾃｰﾌﾞﾙ --------
tempo_table:
        addwf   PCL,F
        retlw   229                     ; 80 BPM
        retlw   183                     ; 100 BPM
        retlw   152                     ; 120 BPM
        retlw   130                     ; 140 BPM

        ;-------- 0〜3に応じてｽﾃｯﾌﾟ数設定用の数値を返すﾃｰﾌﾞﾙ --------
step_table:
        addwf   PCL,F
        retlw   8                       ; 8 ｽﾃｯﾌﾟ
        retlw   10                      ; 10 ｽﾃｯﾌﾟ
        retlw   12                      ; 12 ｽﾃｯﾌﾟ
        retlw   16                      ; 14 ｽﾃｯﾌﾟ

;=========================================================================
check_all_off:
        ;-------- check all off ---------
        movf    rhythm_buf,W            ; rhythm_bufの2ﾊﾞｲﾄが
        iorwf   rhythm_buf+1,W          ;       すべて0なら以下へ
        btfss   STATUS,Z                ;
        return                          ; そうでなければﾘﾀｰﾝ

        bcf     INTCON,GIE              ; ｳｪｲｸｱｯﾌﾟ時に割り込みﾙｰﾁﾝへ
                                        ;       飛ばないように
                                        ;       割り込みを禁止しておく

        bsf     INTCON,RBIE             ; PORTBの変化でｳｪｲｸｱｯﾌﾟを
                                        ;       起こすように設定

        sleep                           ; ｽﾘｰﾌﾟ...

        bcf     INTCON,RBIE             ; 割り込み条件を元に戻す
        bsf     INTCON,GIE              ;         //

        movf    count0_init,W           ; count0をﾘｾｯﾄ
        movwf   count0                  ;

        bsf     event,TRIG              ; すぐに後半処理を行うように
        bsf     stat,HALF               ;       ﾌﾗｸﾞを立てておく

        return

;=========================================================================
        end
```



第7回 CGIスクリプトのセキュリティについて考えよう

横井　勝 <masaru@turbolinux.co.jp>

前回は，CGIの簡単なプログラムを紹介しました。CGIを用いれば，プログラムを介した非常に変化に富んだWebサイトを構築できますが，その反面プログラムを実行しているために，そのプログラムのバグが損害を与える場合もあります。今回は，CGIスクリプトを作るうえで配慮すべきことを紹介します。

ここで紹介する問題点の事例を実際のCGIで利用すると，その問題が発生してしまうかもしれません。実際に稼働中のシステムでは絶対に実験したり悪用しないでください。

データの漏曳防止

CGIに限った問題ではないのですが，インターネットなどの外部に公開するシステムでは，データの漏曳に対して配慮をする必要があります。このデータの漏曳への対策を怠ったせいで，企業の顧客リストが外部に漏曳してしまったという事件などが多くあります。

1.1. 掲示板のファイルの指定方法を用いた問題

たとえば，並列した複数のテーマごとにスレッドといわれる形式にメッセージを書き込んでいくBBSの場合，テーマごとにファイルを作成して，メッセージを閲覧するときにはそのファイルを開くようになっている場合があります。

そして，そのメッセージのファイルは，メッセージの番号に対応するように数値を含んだ形式のファイルになっていると仮定します。

第1図のような場合，メッセージを読むための機能をPerlでいい加減に書くと，**第2図**のようになります。

一見，とくに問題なさそうですが，じつは大きな問題があるのです。一般的に利用している間は，普通に指定されたx番目のメッセージを表示するのですが，仮に悪意のあるユーザーがメッセージの番号に指定する文字列に，わざと実在するファイルへの相対パスを指定すると，他のファイルが閲覧できる可能性があるのです。

たとえば，この環境で，「**http://hogehoge/bbs.cgi?mes=../bbs.cgi**」というふうに，URLをWebブラウザで指定すると，このBBSのプログラム自体を読み込んでしまうのです。この方法を用いて，外部に漏れては困るようなファイルを読まれてしまったと考えてみてください。非常に深刻な問題であると理解していただけるでしょう。

1.2. 対策方法

これは単に指定されるメッセージを，指定されたがままに表示するようになっているのが原因なのです。

このBBSでは，メッセージの番号を用いてメッセージを指定する方法になっているので，「mes=??」で指定される内容は，本来は一つ以上の0から9までの数値の文字列だけになっているはずです。それ以外の文字列が指定された場合は，エラーになるように文字列を検査すれば大丈夫です。この検査をPerlの正規表現を用いてプログラムに追加してみます（**第3図**）。

ここでは「**/^[0-9]+$/**」という正規表現を用いています。この正規表現のなかの「**^**」の部分は「この後に指定する

```
public_html/bbs/ ← BBS のディレクトリ
  + bbs.cgi      ← BBS のプログラム
  + data/        ← メッセージを保管する
                   ディレクトリ
    + 1          ← メッセージ番号 1 の
                   メッセージ
    + 2          ← メッセージ番号 2 の
                   メッセージ
    + 3          ← メッセージ番号 3 の
                   メッセージ
    + 4          ← メッセージ番号 4 の
                   メッセージ
    .....          (以下省略)
```

≪第1図≫数値を含んだ形式のメッセージ・ファイル

```
my $Query;

# CGI の実行時の引数で a=b の形式で指定された文字列を
# $Query{a} = b の形式に保存します。

foreach my $each (split(/&/, $ENV{'QUERY_STRING'})) {
  my ($name, $val);
  ($name, $val) = split(/=/, $each);
  $Query{$name} = $val;
}

print "Content-type: text/html; charset=EUC_JP¥n¥n";

# CGI の実行時に num=?? で指定されたメッセージを表示
# するようにします。

if(open(IN, "./data/$Query{'mes'}")) {
  # ファイルの Open に成功したら読み込みを実行
  while(my $line = <IN>) {
    print $line;
    #
  }
} else {
  # ファイルの Open に失敗したときの動作をここに書く。
}
```

≪第2図≫メッセージを読むための機能をPerlで書いた例

```
# /^[0-9]+$/ という部分は正規表現になります。

 if ($Query{'num'} !~ /^[0-9]+$/) {
   # エラーのための処理を実行
   &Error();
 } else {
   if (open(IN, "./data/$Query{'num'}")) {

   .... 省略 ....
```

≪第3図≫文字列の検査をPerlの正規表現を用いて追加したプログラム例

文字列ではじまって」という意味で，「**[0-9]**」の部分は「0から9までの文字列が該当する」という意味です。また「**+**」の部分は「この前に指定された文字が1文字以上続く」という意味で，「**$**」の部分は「この前に指定された文字で終わる」という意味になります。

そしてこの正規表現をまとめると，「0から9までの文字で始まり，0から9までの文字が1文字以上続いて，0から9までの文字で終わっている場合は，真(True)を返す」という意味になります。

簡単に表現すると「0から9までの文字のみが含まれている場合は真(True)を返す」という意味です。

そして，この正規表現がマッチしない，つまりTrueでない場合には，エラーのときのコードを実行するようになっています。

このようにすれば，数値以外の文字列を指定された場合はエラーとなり，結果として「**../**」といった相対パスを指定する文字列を用いて，他のファイルを閲覧することができなくなります。

予定外な外部のコマンドを実行できてしまう問題

スクリプトの書きかたによっては，想定していない外部のコマンドを実行できてしまうという問題が発生したりします。

外部のコマンドが実行できるということは，ファイルなどの情報の持ち出し，削除，書き換え(改竄)ができるばかりでなく，裏からの進入路であるバックドアを設置されたりすることもあるでしょう。とにかく何でもされてしまうので要注意です。

2.1. 外部コマンドを実行するときに注意

CGIのプログラムからメールを送る機能を付ける場合などでは，外部のプログラムであるsendmailを実行する場合があります。場合によっては，この外部のプログラムを実行する部分を悪用して，任意のコマンドを実行させることも可能です。

たとえば，CGIの実行時の引数で指定されたメール・アドレスにメールを送るように，いい加減にプログラムを書くと**第4図**のようになるでしょう。

これも先ほどと同じく一見普通に見えてしまうのですが，仮にメール・アドレスの後ろの部分に「**;**」や「**&&**」をつけてその後に任意のコマンドを指定してみると，そのコマンドが実行されてしまいます。

この場合の原因は，Perlのopen関数の仕様にあります。open関数で「|」を用いて標準入力や標準出力を受け渡しするためのパイプコマンドを用いると，その部分は「**/bin/sh**」などのシェルを

介して実行されるようになっています。

たとえば，「$Query{'mailto'}」の中身が「masaru@turbolinux.co.jp ; true」となっている場合，ここでは「/usr/sbin/sendmail masaru@turbolinux.co.jp ; true」というコマンドが実行されてしまうのです。

2.2.対策方法

この場合も，先ほどと同じくCGIの引数を検査するという方法があります。もちろん，メール・アドレスに利用できる文字のみが指定されているかどうかを検査する方法もありますが，シェルで他のコマンドを実行できてしまう文字を避けるように正規表現を設定する方法もあります。

たとえば，bashなどのBourne shellでは，"&","&&","(",")",";;",";","|","||"などの文字と改行が制御演算子（Control Operator）となっており，"<",">","<<",">>","<>","<&",">&","<<-",">|"の文字がリダイレクト演算子（Redirection operator）に用いられています。そしてこれらの文字を用いると，特定のコマンドを実行したり，その実行結果を特定のファイルに書き込むことができます。くわしくは，コマンドman shで閲覧できるマニュアルをご覧ください。

これらの文字を排除するように正規表現を書くのですが，もう少し簡単に考えて，改行（0x0a）,"&"（0x26）,";"（0x3b）,"|"（0x7c）,"("（0x28）,")"（0x29）,"<"（0x3c）,">"（0x3e）の文字に反応する正規表現を書いてみます。でも，これらの文字は正規表現の特殊な意味をもつ文字列でもあるので，ここでは見やすくするために，16進数を直接指定するようにして書きます（**第5図**）。

このようにすれば，確実に制御演算子やリダイレクト演算子を排除することができるでしょう。

しかし，場合によっては，これらのシェルでの特殊な文字をプログラムの引数として実行しなければならない場合もあります。このような場合は，引数をシングルクオート（'）で囲って指定された文字列の中にシングルクオートがないように正規表現を設定したり，バックスラッシュ（¥）を活用するなどの方法があります。

セキュリティ対策に完璧は存在しない

今回は二つの事例を紹介しましたが，CGIのセキュリティ対策というのはもっといろいろな方法があります。そして，攻撃する方法は日夜研究開発されており，それを防御する方法も日々進化しています。その状況は，ほとんど「イタチゴッコ」だといっても過言ではないでしょう。

対策するうえで，とっても重要なことがあります。それは，どこかに危険が潜んでいるという意識を持つことです。コンピュータのプログラムは，構造が難しければ難しいほど完璧なものになりにくくなってしまいます。

しかし，完璧でないものを少しでも良いものにする努力は必要です。これはプログラムを書く人たちだけでなく，それを利用する人たちにも知っていただきたいことなのです。

```
foreach my $each (split(/&/, $ENV{'QUERY_STRING'})) {
   my ($name, $val);
   ($name, $val) = split(/=/, $each);
   $Query{$name} = $val;
 }

 # mailto=?? で指定されたメール・アドレスにメールを送る。

 if(open(SENDMAIL, "| /usr/sbin/sendmail $Query{'mail
 to'}")) {
   print SENDMAIL "From: BBS <BBS>¥n";
   print SENDMAIL "To: $Query{'mailto'}¥n";
   print SENDMAIL "Subject: From BBS¥n¥n";
   # メールの本文の送信
   print SENDMAIL $Honbun;
   close(SENDMAIL);
 }
```

≪第4図≫CGIの実行時の引数で指定されたメール・アドレスにメールを送るプログラム例

```
if ($Query{'mailto'} =~ /¥x0a|¥x3b|¥x7c|¥x28|¥x29|¥x3
c|¥x3e/) {
   # エラーの時のプログラムを実行
   &Error();
 } else {
   if(open(SENDMAIL, "| /usr/sbin/sendmail $Query{'ma
   ilto'}")) {
     print SENDMAIL "From: BBS <BBS>¥n";
     print SENDMAIL "To: $Query{'mailto'}¥n";

   .... 省略 ....
```

≪第5図≫特定の文字に反応する正規表現例（16進数を直接指定）

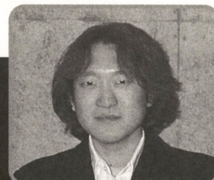
DTM Art Studio
パソコンで音楽をたのしもう!

協　力
ヤマハ（http://www.yamaha.co.jp）

Text&SoundData
仲野方敏／上村学

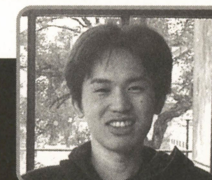
このコーナーは，芸大生が体験したDTMの基本，応用，実践を紹介するコーナーです

筆者紹介



仲野方敏：
大阪芸大
芸術学部　音楽学科
音楽工学コース（音楽工学専攻）3回生

みなさん，こんにちは。本格的な夏も終わり「夏休みボケで作曲する気にならない」なんて言ってる場合じゃありません。音楽家に休む暇などありません。極限状態でこそ，新たなイマジネーションが生まれるのです。ということで，今月もがんばっていきましょう。



上村学：
大阪芸大
芸術学部　音楽学科
音楽制作コース（作曲専攻）3回生

先月号から始まった「らくらく作曲名人2」の連載ですが，みなさん「らくらく作曲名人2」の印象はどうでしょうか？「めちゃらく作曲名人」より難しそうですが，できることも増えて自由に作曲やアレンジが可能なので楽しそうですよね。今月も楽しくがんばりましょう。

YAMAHA「らくらく作曲名人2」～アレンジ編～

先月号のおさらい

前回は，まず「らくらく作曲名人2」ソフトとソフトシンセサイザのインストール方法の説明と設定をしました。

インストールに関しては，画面の指示どおりに進めれば問題ありません。設定は，ソフトを起動して[設定]メニューから[音源設定]を選択し，「ソフトシンセサイザを使用する場合」をクリックして[詳細設定]をクリック。その後，MIDI Outを「YAMAHA SXG Driver」にするんでしたね。

それから「めちゃらく作曲名人」で作ったデータの簡単なEDIT方法を説明しましたね。音色（ボイス），残響（ホールタイプ），スタイル，テンポのEDITをしましたが，わかりましたか？その後はさらに難しかったかもしれないですけど，楽譜に音符を入力する方法を説明しました。初心者には難しかったかもしれないですが，音符入力ができるようになったら他の音楽ソフトにも応用が利きますし，「らくらく作曲名人2」でもできることが広がりますので，少しずつなれましょう。

最後に今月号の予習みたいな感じで「らくらく作曲名人2」の楽しい使いかたと題して，メロディを楽譜入力してアレンジする方法を説明しました。

さぁ，それでは今回はさらに進んだアレンジにチャレンジです！！

「らくらく作曲名人2」

音楽の専門知識がなくても，作曲・編曲等が簡単にできるソフトです。「めちゃらく作曲名人」とちがい，シーケンサとして使うこともできるので，DTM初心者以外にもお勧めです。それに，楽譜を制作することも可能です。

製品仕様
●CD-ROM（「らくらく作曲名人2」，ソフトシンセサイザ「S-YXG50」，ヤマハCBXドライバ，MidRadioプレーヤ，用語集）
●取扱説明書

動作環境
●対応パソコン：NEC PC-98シリーズ，IBM-PC／AT並びにその完全な互換機
●対応OS：WindowsMe/98/95/NT4.0/2000
●CPU：Pentium-166MHz以上/Celeron以上（ソフトシンセサイザ同時使用の場合MMX Pentium-233MHz／Celeron以上推奨，Windows2000の場合PentiumⅡ300MHz以上推奨）
●必要メモリ：32MB以上（64MB以上推奨）
●ハードディスク必要容量：40MB以上
●ディスプレイ：800×600ドット，24ビットTrue Color以上
●音源：ソフトシンセサイザまたはXG対応MIDI音源機器
●必要機器：マウス，ハードディスク，2倍速以上のCD-ROMドライブ



14,800円

モチーフをアレンジしよう

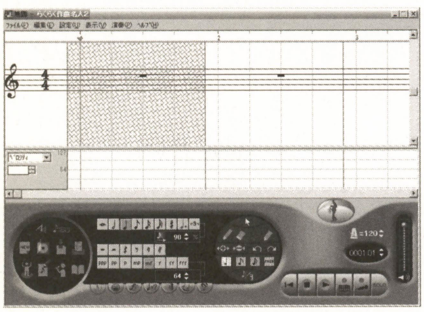
まず，先月号で説明した楽譜入力機

能を使ってアレンジするメロディ（モチーフ）を入力します。今回のメロディ（モチーフ）は，筆者が作曲したSilentDimensionという曲を使います。

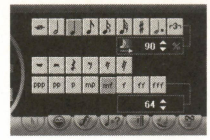
先月号で説明した楽譜入力がまだできない人や難しいという人は，Webのほうにメロディ（モチーフ）のテンプレートがありますので，それをダウンロードしてもらえば，音符入力作業は省くことができます。

音符の入力

まず，ステージ画面で足元の番号が1のプレーヤをダブルクリックして楽譜（スタッフ）画面を開きます。
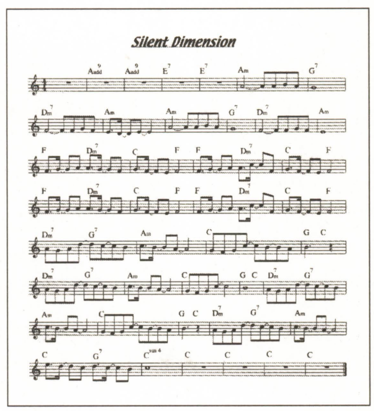


音符パレットの中から入力したい音符の長さと音の強さを選びます。





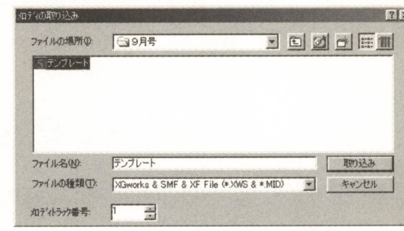
そして，鉛筆ツールを選んで紙面の楽譜を見ながら音符を入力していきます。

音符の長さや音の高さをまちがえないように入力しましょう。また，音の強さをメロディの盛り上がりや自分のセンスで変えると，メロディに表情を付けることができるます。慣れた人は挑戦してみてください。



テンプレートを使う場合

音符入力がまだできないという人は，こちらをお読みください。まず，Webからテンプレートのファイルをダウンロードしておきます。ステージ画面のままで［ファイルメニュー］から［メロディの取り込み］をクリックします。



するとメロディ取り込み画面（ダイアログ）が表示されます。ここでWebからダウンロードしておいたテンプレートのファイルを選択します。［メロディトラック番号］が1になっていることを確認して［取り込み］をクリックします。すると，ステージ画面の足元の番号が1のプレーヤにメロディ（モチーフ）が読み込まれます。また，他のプレーヤは楽屋に待機されます。

これで，音符入力が終わった状態と同じになります。

民族音楽のスタイル

東南アジアの音楽

東南アジアには何百という民族が存在しています。それぞれの民族で文化，習慣，宗教なども異なるということは珍しくありません。このことから民族音楽にもさまざまな種類が存在しています。

そのなかでも，東南アジアの独特な音楽文化の代表的なものといえば，ゴング（銅鑼）やメタロフォン（鉄琴や青銅琴）を用いた合奏音楽です。東南アジアほど，このような楽器の種類の多い地域は世界にはありません。なかでも，とくに重要とされている楽器がゴングです。「ゴングgong」という言葉はそもそもマレー語起源の言葉です。

東南アジアのゴングの形はたくさんあり，直径が1mある吊りゴングや，壷を伏せたような形のもの，中央に突起のあるものなどさまざまです。なかには複数のゴングを一定の音階に調律して，セットで用いる「ゴングチャイム」と呼ばれるものもあります。この楽器は旋律打楽器として，東南アジアの器楽で多用されます。最近は日本でも有名になってきたインドネシアやマレーの器楽合奏のガムランでも使われています。またタイのピーパート，カンボジアのプンピアット，ミャンマーのサインワインなどにもゴングはかかせません。

そしてもうひとつ，東南アジア独特の旋律打楽器メタロフォンは，鉄や青銅でできた音板の下に竹の共鳴筒をつけたものと，音板を共鳴箱にのせたものがあり，主としてインドネシアのガムランで用いられます。ガムランgamelanとは，「ガムルgamel」すなわち，「たたく」から派生した言葉であり，「たたいて鳴らす楽器」を意味します。大編成のガムランでは70～80の楽器が並びます。大型の吊りゴングは後方に置かれ，メタロフォン，ゴングは真ん中に置かれます。そして音量の小さいルバブという弦楽器（胡弓）やスリン（縦笛）などは前のほうにすわります。全体の中央には太鼓（クンダン）が置かれて，太鼓奏者が，テンポ，緩急，音量などの変化を微妙な打ちかたで合図します。

また，ガムランの音階は2種類あり，洋楽の長音階や短音階とはまったくちがいます。オクターブをほぼ5等分したスレンドロ音階とオクターブを異なる間隔で7分したうちの5音を使うペロック音階があり，5音音階で演奏されます。そして，演奏形式はコロトミー構造といい，一定の拍子の区切りを繰り返す形式をもっています。この構造は，広く東南アジアのほかの器楽合奏にも共通して見ることができます。東南アジアは竹の産地でもあり，竹製の楽器も無数にあります。

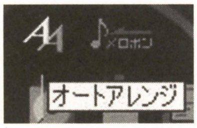
このように東南アジアの音楽はゴングやメタロフォンや竹製の楽器を中心に使った合奏の音楽がたくさんあります。最近では，CD屋さんの民族音楽コーナーにガムラン等のCDも置いているので，興味を持った人は聴いてみてはいかがでしょうか？いろんな楽器が一定の周期をきざんでいて不思議な雰囲気を味わえます。

このような構造の音楽（音階は除く）ならば簡単にDTMでも作れるので，新しい音楽感が生まれるかもしれませんよ。

アレンジする

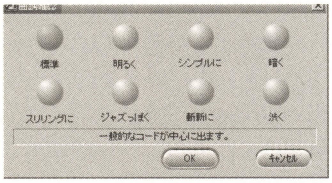


入力が終わったらステージ画面へ戻ります。画面左側のカーテンの上にマウス・ポインタを移動すると，スタイルパレットが開きます。いずれかのアイコンを選ぶとそのスタイルが有効となります。



ここまでの作業が終了したらオートアレンジ機能を実行します。ステージ左下のボタンをクリックすると曲調確認画面が表示されます。

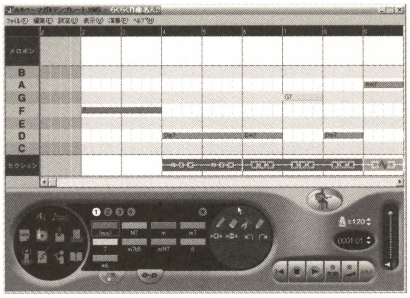
曲調ボタンの中からいずれかをクリックして選ぶと，その曲調に関する簡単な説明が画面の下に表示されます。［OK］ボタンをクリックすると，選択した曲調の伴奏が自動生成されます。



アレンジデータの修正

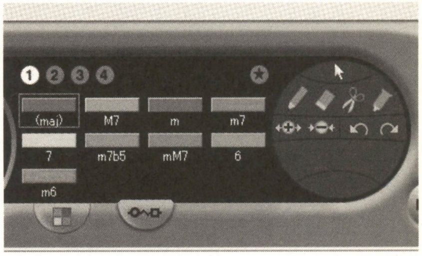
アレンジ画面を使って，コードの一部を変更するなどのアレンジデータの修正／編集を行うことができます。アレンジデータは大まかにコードとセクションに区別されます。ハーモニーが気に入らない場合はコードを，曲自体の構成が気に入らない場合はセクションを修正します。

スタイル用のプレーヤ（足元が赤）をダブルクリックすると，アレンジ画面が表示されます。



コードの修正

アレンジ画面内の上半分にコード名つきの四角形が段違いでいくつか配置されています。これら一つひとつがコードを示しています。表示されている段でルート［根音］を，色でタイプを表します。コードの修正は，この四角形をマウス操作で動かしたり，画面中央下のコードパレットを使って行います（ルート［根音］とは，和音［コード］の基礎となる音で，和音［コード］はこのルート［根音］の上に音を積み重ねることによって成り立っています）。



画面中央下のパレット下部のタブをクリックすることで，コードパレットが表示されます。パレットの上部1～4のボタンでコードタイプのページを切り替えます。また，右端の星のマークをクリックすると，コード候補（ある場合）のページを開くことができます。

この，コード候補のページは，差し替えられる（お勧めの）コードの候補を表示します。この機能を使うことで，あまりコードにくわしくない人でも簡単にハーモニーを変更することが可能です。



お勧めコードと差し替えるには，まず，画面右下のツールボタンから［矢印（アロー）ツール］ボタンをクリックして，矢印（アロー）ツールを選びます。

つぎに，差し替えたいコードをクリックして選びます。選ばれたコードは赤くなります。

コードパレットで星のマークが点灯して，コード候補がアイコンとして表示されます（コード候補がない場合は，

シリーズ音響学

今回の「シリーズ音響学」では，音に関する基礎的な知識を説明したいと思います。

この世に存在する音には，さまざまなものがあります。前回でも述べたように，音が空気の振動である以上，それを目で見てちがいを識別することは不可能です。そこで，スタジオや音響分析の現場では，オシロスコープやスペクトル・アナライザと呼ばれる測定器を用いて，音をグラフとして視覚化し，そのちがいなどを識別しています。

コンピュータのシーケンス・ソフトや音楽制作に使われる機材の説明書には，上記の測定器によって得られたグラフや専門用語が書かれています。ここでは，それら専門用語について説明してみたいと思います。

波形／周波数／位相／音圧／スペクトル

●波形：音圧の変化を時間軸に沿って表したもの。

●周期：周期的な現象において，同一の状態が再現するまでに経過する最小時間間隔。

●波長：波形の基本パターン1周期分の長さ（垂直距離）。

●周波数：周期的な現象が毎秒繰り返される回数。Hz（ヘルツ）という単位で表される。ちなみに，440Hz（ラの音）というのは，1秒間に空気が440回振動しているということ。

●位相：波形の1周期を360°として，ある点からの位置を角度で表す。これを位相（位相角）と呼ぶ。位相のズレや遅れも，その量を位相角で表す。たとえば半周期分位相が遅れた場合には，180°の位相遅れと表現される。ちなみに，位相が180°ズレた音（逆相）をもとの音とミックスすると，音が消える（相殺される）。これは，音波の疎（－）の部分と密（＋）の部分が打ち消し合ってしまうため。また，私たち人間は位相のちがいをほとんど識別することができない。

●音圧：音が存在すると，媒質（空気など）の圧力が変化する。音圧とは，圧力の平均値からの変化部分をいう。

●スペクトル：複合音を純音に分解して，グラフ（関数）として表したもの。どんな周波数の音がどれ位の強さでその複合

表示されません）。これらの，アイコンをクリックするとコードが差し替わります。コード候補の響きを聴いてみたい場合は，候補アイコンをマウス右ボタンでクリックして［試聴（M）...］を選んでください。

また，誌面の楽譜の上にコードが書いてありますが，筆者が作曲するときに付けたものなので，そのとおりにする必要はありません。参考に見る程度でいいです。

音楽知識のある人は…

コードのルートを変える

音楽の知識のある人は，自分で考えたコード入力をすることができます。コードのルート（根音）を変える場合は，マウスポインタを変えたいコードの上に移動して，マウスポインタが上下の矢印になったら上下にドラッグします。画面の左側に下からCDEF...のアルファベットが並んでいます。これらが各コードのルートを表しています。半音を設定したいときはアルファベットの間にコードを配置します。

コードのタイプを変える

［矢印（アロー）ツール］ボタンを選んで差し替えたいコードをクリックして選びます。その後，コードパレットの1〜4ページの中から，目的のコードタイプアイコンをクリックします。すると選んでいたコードの色が変わると同時に，コードのタイプが変わります。

アレンジ画面のコードを消去する



アレンジ画面上に配置されたコードを消去するには，ツールボタンから［消しゴムツール］ボタンをクリックします。消去したいコードの上を消しゴムツールでクリックすると，そのコードが消去されます。

複数のコードに分割する

あるコードを複数のコードに分割することもできます。分割したコードを別のコードタイプに差し替えることにより，現在より細かい単位でコードが変化する曲に修正できます。ツールボタンから［はさみツール］ボタンを選択して，分割したいコードの分割したい場所でクリックするとその場所でコードは分割されます。



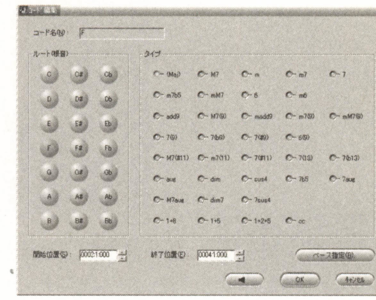
複数のコードを結合

逆に，複数のコードを一つに結合することも可能です。ツールボタンから［ボンドツール］ボタンを選びます。結合したい場所をボンドツールでクリックすると，コードが結合されます。複数のコードの位置を一度に変えたいときに便利です。



視聴する

修正したコードを試聴するには，試聴したいコードを選んで［詳細（D）...］を選びます。するとコード編集画面が表示されます。

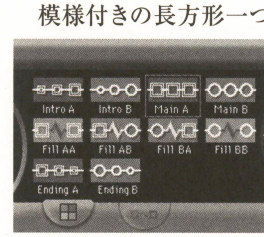


画面の下にあるスピーカのイラストのボタンをクリックすると，試聴できます。また，この画面でコードタイプやルートを選んで編集することもできます。画面の下にある［OK］をクリックすると，コード編集画面が閉じます。

セクションの修正

アレンジ画面の一番下（パレットの下）の右側のボタンをクリックするとセクションパレットが表示されます。

模様付きの長方形一つ一つがセクションを表しています。セクションとは，曲を構成するパターンを意味して



音中に存在しているかということが示されている。

●**周波数成分**：スペクトルを形成する各々の周波数の音。

●**周波数帯域**：周波数成分が分布している周波数の範囲。

音の種類

●**純音**：ただひとつの周波数のみで構成された音で，波形は正弦波（サイン波）。ラジオなどで流れる時報の「ポッ，ポッ，ポッ，ピー」という音がそれ。数学的に最も単純という意味で純音と呼ばれる。理論上では，純音を組み合わせることによってどんな波形でも作り出すことが可能とされている。シンセサイザの基本的な原理はこの理論に基づいている。

●**複合音**：複合音には，「調波複合音」と「非調波複合音」の二つが存在する。単に，複合音といった場合には，複数の周波数の音によって構成された音（純音以外の音）全般を指す。

●**調波複合音**：基本音と倍音によって構成された音で，楽器などの音はこちらの部類に入る。

●**非調波複合音**：基本音と倍音という関係が成り立っていないもの。「サー」というノイズなどがこれ。

●**基本音**：調波複合音の場合は，その周期に相当する振動数（周波数）の純音。非調波複合音の場合は，その成分中振動数が最小の（最も低い）もの。

●**基本周波数**：基本音の周波数。

●**倍音**：基本音より高い周波数成分が基本周波数の整数倍である場合に，その成分を「倍音」，その正弦波を「高調波」と呼ぶ。特定の倍音を指す場合には，「第n（2，3…）倍音」というふうに呼ぶ（第「1」倍音という呼びかたは，まず使わない。第1倍音というと，数学的には基本音を指すことになる）。

●**雑音**：スペクトルの周波数成分や振幅が時間的に不規則に変化し，一定のスペクトルを持たない音（周期性がなく，部分的に分解できない）。

補　足

●**オシロスコープ**：「波形」の観測に使用する装置（理科や物理の教科書に写真が載っていると思うので，そちらで確認してください）。

●**スペクトル・アナライザ**：音を構成している周波数成分と振幅との関係を分析し，棒グラフのように表示する測定器。

います。たとえば,「イントロ」とか「サビ」「Aメロ」などの言葉を聞いたことがある人も多いのではないでしょうか？

「イントロ」とは曲の出だしの前奏部分をいいます。「サビ」とは曲のメロディの一番盛り上がる部分をいいます。メロディの中で出てくる一番高い音は，普通この部分で使われる場合が多いようです。「Aメロ」とか「Bメロ」は，サビ以外のメロディのことをいいます。

「らくらく作曲名人2」では，セクションを「イントロ」,「メインA」,「メインB」,「フィル」,「エンディング」にわけています。

「メインA」「メインB」とは，曲の中で使われるメインパターン，つまり「Aメロ」や「Bメロ」や「サビ」で使えることを表します。

「フィル」とは，曲の雰囲気を変えたり，メリハリを付けるときに「メインA」や「メインB」の間に挟んで使います。「フィルAA」なら「メインA」と「メインA」の間に挟んだり，「フィルAB」なら「メインA」と「メインB」の間に挟んで「メインA」のセクションから「メインB」のセクションにスムーズに入れるようにして使います。また，フィルのAやBを無視して，「メインA」と「メインA」の間に「フィルBB」を挟んでみるのもおもしろいかもしれませんね。

そして「エンディング」とは，その名のとおり，曲の最後の部分のセクションですので，メロディが終わったところから置くのが普通です。

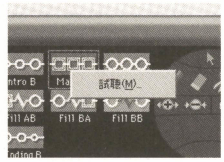
セクションのタイプ変更

それでは，セクションのタイプを変えてみましょう。まず，ツールボタンの中から［矢印（アロー）ツール］ボタンをクリックして，マウスポインタを矢印（アロー）ツールにします。そして，画面上半分にあるコード表示領域のすぐ下にセクションを表示している部分がありますから，差し替えたいセクションをクリックして選びます。



選択されたセクションは赤くなります（編集できるセクションの最小単位は1小節ですが，オートアレンジ機能を使っているので，複数小節連続してつながって表示されている場合もあります）。

セクションパレットの中から，変えたいセクションタイプのアイコンをクリックします。すると選んでおいたセクションのタイプが変わります。セクションタイプを試聴したい場合は，セクションタイプアイコンをマウスの右ボタンでクリックして，［試聴］をクリックしてください。



セクション位置の変更

セクションの位置を変えたい場合には，セクションを左右にドラッグして，置きたい場所に配置します。1小節単位での移動ができます。なお，すでにセクションがある場所にも重ねて移動することができます。この場合，上に置かれているセクションが有効になりますので気を付けてください。

セクションの消去

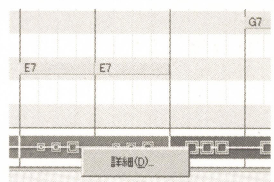
セクションを消去したい場合は，コード消去と同様の操作で行うことができます。消しゴムツールをツールボタンから選び，消去したいセクションの上で消しゴムツールをクリックします。

セクションの分割／結合

セクションの分割や結合に関しても，コードを修正する場合と同じ操作で行うことが可能です。分割する場合は，はさみツールを分割したい場所でクリックします。結合する場合は，ボンドツールを結合したい場所でクリックします。

視聴

修正したセクションを試聴するには，試聴したいセクションをクリックして選択状態（赤色）にし，マウスの右ボタンをクリックして［詳細（D）...］を選びます。セクション編集画面が表示されます。



セクション編集画面の下にある［試聴（M）］ボタンをクリックすると試聴することができます。

また，別のセクションタイプを選んで編集することも可能です。画面の下にある［OK］または［キャンセル］ボタンをクリックすると，セクション編集画面が閉じます。

今月の復習

今回はメロディ（モチーフ）を入力して，そのメロディに伴奏をつけてコード等を編集したわけなんですが，どうだったでしょうか？少しずつ難しくなっていますよね。でもその分，曲の自由度は高くなって，おもしろかったんじゃないでしょうか？まだコードとかわからないという人でも，セクションの編集は簡単なので，それだけ変えてみても十分個性的で，おもしろかったんじゃないかと思います。

お手本として，筆者がアレンジしたデータをホームページに掲載しますので参考にしてみてください。

次回は「らくらく作曲名人2」で作曲＆さらに進んだアレンジに挑戦したいと思います。

◆Webの内容
- モチーフテンプレート
- 芸大生のモチーフアレンジデータ
- コンテストのくわしい応募方法

◆次回予告
- 「らくらく作曲名人2」アレンジ編②
- 南アジアの音楽（コラム）
- シリーズ音響学（コラム）

「らくらく作曲名人2」作品募集！

前回募集した「めちゃらく作曲名人」で作ったデータを「らくらく作曲名人2」でEDITしたデータの発表会がWeb上で行われています。みなさんのEDITしたデータを聴くとさまざまでおもしろいですね。今回も「らくらく作曲名人2」で作った作品発表会の曲データを募集します。

今回は，今月号で使ったモチーフ（メロディ）のアレンジデータを募集します。作品の発表会は，来月のWeb上で開催しますので，どしどし応募してください。

参加希望者は以下のデータを用意し，「らくらく作曲名人2・作品募集」係まで郵送か，Eメールでお送りください。

・応募する際，用意するもの

①モチーフアレンジ・データ，MIDI・xwsファイル
②作品の簡単な紹介，コメント
③作曲者プロフィール
④連絡先（Eメール・アドレス）
（郵送の場合は，フロッピーディスクで）

http://www.basicmagazine.net
http://www.beeplus.or.jp/bm/art/





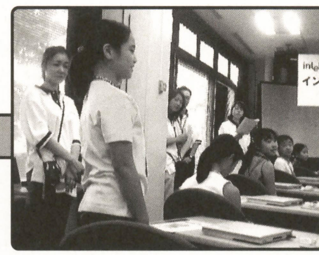
電脳キッズ，田園に大集合！

インテル夏休みPCスクール 同行記

ザリガニやトンボを捕まえて，デジタルに観察・記録しよう!!



▲インテルつくば本社の1階トレーニングルームに設けられたPCスクールの教室

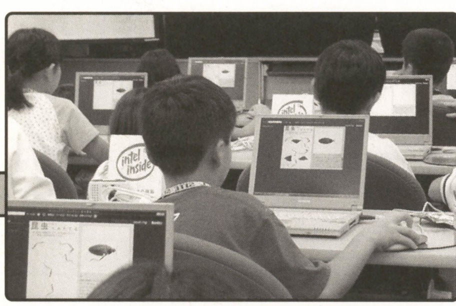


▲はじめにみんなの自己紹介から。学年と夏休みの楽しかった思い出を発表。「東京へ行った」「ハワイアンズ※へ行った」など家族旅行の思い出が多かった

問題1.パソコンは何が進化したものでしょう？
① テレビ
② 電卓
③ タイプライタ



▲午前中はIA技術本部の阿部本部長によるパソコン講座。スライド・ショーを使ったクイズ形式の問題を中心に，マザーボードやCPUを実際に手に取って，それらについての解説を行った



▲Microsoft Encarta（マイクロソフト・エンカルタ）を使って“調べ学習”のトレーニング。このページではバラバラになった昆虫を組み立てられる。正しく組めると昆虫は，羽根を広げて飛んでいってしまう。かなりリアルで，ちょっと恐かった？



▲ランチタイムは社員のみなさんも利用している大食堂で，みんなでワイワイ食べました





※**ハワイアンズ**…『スパリゾート・ハワイアンズ（http://www.hawaiians.co.jp/）』福島県にある温泉テーマパーク。インテルのスタッフにはナゼか旧名の『常磐ハワイアンセンター（11年前に改名）』のほうが，とおりが良かったようです。

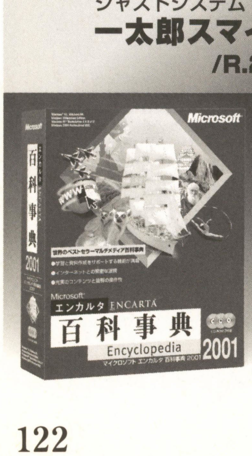
■ソフトウエア

全国1万校で採用されている
小学生用ワープロ・ソフト

ジャストシステム
一太郎スマイル
/R.2［家庭学習用］
価格：8,800円

一太郎スマイル /R.2
［学校版］
価格：12,800円





調べ学習の強い味方
マルチメディア学習支援ソフト

マイクロソフト
エンカルタ百科辞典2001
価格：オープン価格（販売価格27,600円）

■周辺機器

パソコンにつないでPCカメラ
単独でデジカメとして使える

Intel Pocket PC Camera
価格：16,800円





ミクロの世界をパソコンで見よう！
デジタル世代の顕微鏡

Intel Play QX3
コンピュータマイクロスコープ
日本語版
価格：14,800円



田んぼは楽しい学校だ

夏休みも終盤に近付いた8月20日の月曜日，小学校4年生から6年生を対象としたパソコンスクール“インテル夏休みPCスクール”が茨城県つくば市のインテル本社で開催されました。

この“インテルPCスクール”は，1999年から毎年春・夏・冬の長期休みに行っている教育支援プログラムの一環です。今回は，つくば市内の小学校に通う小学生およそ100人の応募者のなかから抽選で選ばれた20人が参加しました。

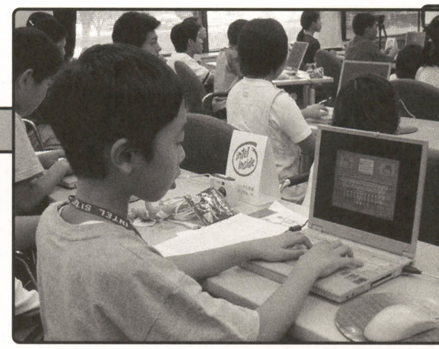
毎回趣向を凝らしたテーマで開催されるPCスクールの今回のテーマは，“田んぼにすむ生き物を観察して，パソコンで絵日記を作成しよう”というもの。実際に野外で生き物を捕まえ，観察するという夏休みらしい研究学習となりました。



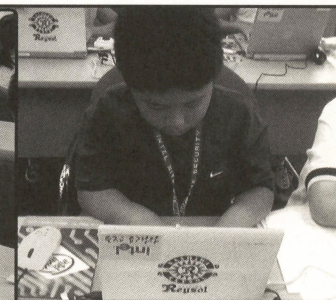
▲はじめて見るパソコンのパーツも，手で触って実感。理論よりも体感から！



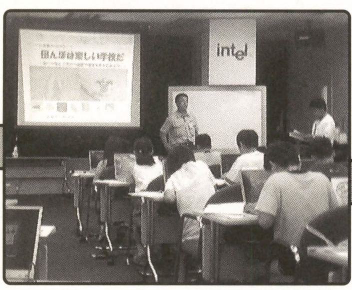
問題2.CPUはシリコン・ウエハーから作られますが，このウエハーの原料は何でしょう？
①プラスチック　②石
③金塊　④ガラス



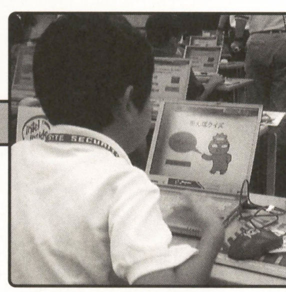
▲続いてはタイピング・ソフトでキー入力レッスン。『キーボード・ファイターII』はゲームしながらキー入力を覚えられる楽しいソフト（「一太郎スマイル」に付属）。みんなでスコアを競い合った



▲子どもたちが実習で使ったパソコンは日立製作所のFLORA 220シリーズ。その何台かは柏レイソル公認モデル『FLORA 11（通称サッカーパソコン）』だった



▲午後は，いよいよ今回のテーマ『田んぼ』の学習。講師の建元先生は教員対象の情報教育支援プログラム『Intel Teach to the Future』の受講者。学習の成果をフルに発揮していました

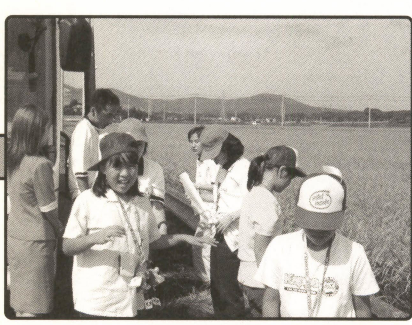


▲田んぼクイズに挑戦だ

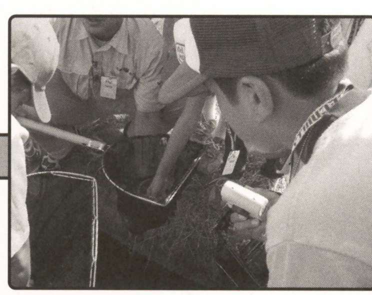
田んぼクイズ
日本全国にあるダムと田んぼをくらべるとどちらのほうがたくさん水をたくわえることができるでしょうか？
①同じくらい
②ダム
③田んぼ



▲『Yahoo!きっず（http://kids.yahoo.co.jp）』を使い，インターネットで「田んぼ・かえる・とんぼ」のキーワードでホームページを検索



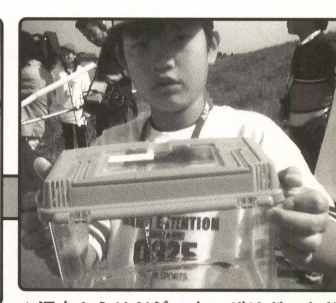
▲バスに乗って田んぼへ移動。外は暑いぞ！



▲用水路に仕掛けた網には，どんな生き物が掛かっているかな？



▲2人1組で『Intel Pocket PC Camera』で捕った生き物を撮影



▲泥水からはドジョウ，ザリガニなど。トンボやコオロギなども採集した（水槽に入っているのは，小さなザリガニ）

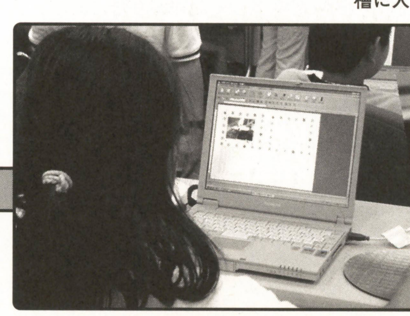


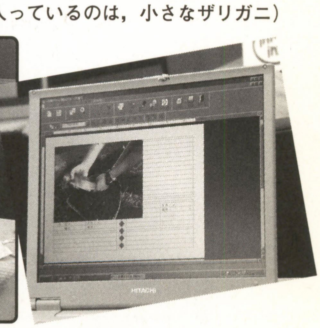
▲採集した生き物を『Intel Play QX3 コンピュータマイクロスコープ』で観察





▲怪獣みたい？





▲田んぼで撮った画像を貼り込み『一太郎スマイル』で絵日記にまとめた。これは後日CD-Rに焼かれ，子どもたちにプレゼントされる

問題1の答え=②電卓　問題2の答え=②石　田んぼクイズの答え=③田んぼ

あくせすROOM

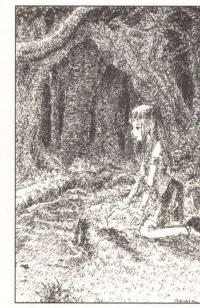


新企画

影：やがて世界は改革されるだろう！
編：いきなりなんですか。
影：萌えプロの投稿が予想外に多くて，予定よりも長つづきしてしまったけれど，そろそろ新しい企画に切り替えようかと思ったのだ。季節の変わり目，衣替えの季節も近いしな。
編：そうですね，イラスト常連がどうしても有利になってしまったので，誰でも気軽に参加できる遊びがほしいです。
影：じゃあ，趣向を変えて，マイナーなゲームキャラの特徴をいくつか挙げて，そこから勝手に想像したイラストを起こすという企画はどうだ。おもしろければオッケーなんで，気軽に参加できそうじゃん。
編：それは，キャラクターをデザインした人にかなり失礼な結果になりそうで……。
つぐ美：あ，似たのあるよ。わたしが小学生のころに「ひまつぶし選手権」ってあったわね。キャンプとかのレクリエーションで，先生が「あらいぐま」とか「タンポポ」って問題を出して，それを資料もなにも見ないで描くの。もともと小学生だし，とんでもない絵ばっかりできあがって楽しかったわ。
影：あったあった。私の地域もそんな名称だったぞ。人類共通か？　じゃあ，試験的に募集してみるかな。

> **うろおぼえ友の会**
> 2002年1月号お題
> 「原動機付き自転車」
> （締切：2001年10月末日）
>
> ・資料や現物を見ないで描きましょう
> ・これがなんなのか人に聞かないでください
> ・うまく描けても賞品はありません

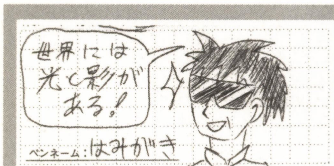
編：なんだか，果てしなく誤解に満ちたイラストがいっぱい届きそうですね。
つぐ美：楽しみ～。



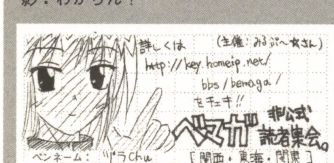
◀北海道札幌市・春陽
影：こ、この描き込みはすごいぞ！
編：誰を想ってユリの花を見つめるのでしょうか。



▶愛知県名古屋市・わいち301
編：ジャズのセッション中と思いきや、じつは星空のプログラマー。夜想曲を紡ぐかのように、芸術的なスクリプトを奏でます



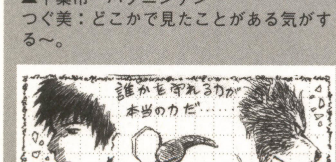
▲北海道天塩市・はみがき
影：わからん！



▲和歌山県橋本市・ヅラChu
影：読者集会焼け？
編：どんどんやっちゃってください。



▲千葉市・バグニンゲン
つぐ美：どこかで見たことがある気がする～。



▲大阪府吹田市・ARAI
つぐ美：大学の話も聞かせてほしいな。

素材名人 萌えプロ

影：発想と発想のぶつかり合い！と書くと聞こえがいいけど，じつは単にネタがかぶってるだけでしたジャンゴー。圧倒的に多いタル！ 樽！ たる！ 西部劇映画が好きな私には，ちょっと物足りなかった気がする内容であったよジャンゴー。というわけで，今月のテーマは

- 妹
- せまいところが好き
- 西部開拓時代

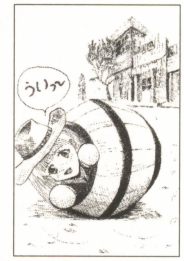
の3件でありましたー。



▶千葉県柏市・鈴木有人
影：映画「スターウォーズ」で、戦闘機の後部座席に乗ったR2D2を思い出したよ。



▶愛知県名古屋市・のうぎょうアッパー
編：帽子に弾丸のあとが！
影：そうとうな石頭だな。
つぐ美：ちょっとは背景ストーリーとか想像してよね！



◀福島市・みなみとらいぱ
つぐ美：1タルまるまる飲み干しちゃったの？
編：人物の各部も、背景なみにしっかり描いてほしかったです。

お題と締切日

影：もうすぐ満期終了の「萌えプロ」。壮大なフィナーレを飾るのは誰だろう？　締切をギリギリまで引き延ばしてみたので，多数の応募を待っているぞ。

11月号
・メイドさん
・楽器が好き
・（自由課題）
（締切：2001年8月30日）

12月号
・防災訓練
・菓子パン
（締切：2001年9月30日）

あくせすROOM応募のルール

●愛読者カードや，官製ハガキ，同サイズの丈夫な紙などでお送りください。

ワンポイント
まとめ送りで大量に送るよりも，気合いの入った数枚の作品のほうが，インパクトは強く，掲載率は上がるようです。

●鉛筆や色鉛筆，涙滴の多いボールペン画は印刷に不向きなので避けましょう。

ワンポイント
下書きのあとが消し切れてないと見苦しくなり，評価が大幅にダウンしてしまいます。仕上げはていねいに。

●投稿内容は，掲載されたプログラムの感想や，学校でのできごと，一発ギャグなど，なんでも結構です。みなさんで楽しいページにしていきましょう。

≪応募先≫
〒141-8715 東京都品川区東五反田1-11-15
電波新聞社マイコンBASICマガジン編集部
あくせすROOM係

webでも受付中です。Internet読者ハガキでどうぞ。
http://www.basicmagazine.net/

大人気クリエイター集団「ギガ連射」があれこれ語る

GIGA RENSYA つつうらら

最終回

ど～もど～も，ギガ連射です。ダイポール現象の影響か，8月現在札幌の天気もさっぱり良くならず，気温も上がらない状態が続いています。日中に外を半袖でいられる日数も去年より少ないですが，そろそろ暑くなってくれるでしょうか？それともこのまま秋に突入？それじゃあまりにも悲しすぎる!!

『会議の旅　in　大倉山』

週に一度の会議の日。いつも会議場所をどこにするかですごく悩むんですが，久しぶりに天気が良く，気温も高いので，気分の良いところで会議しようってことになりました。海とか広い公園などの意見がでたんですが「なんとなく山かな～？」ってことで，札幌近郊の大倉山で会議することにしました。

たまには「リフレッシュも大事だよね？」なんてメンバー同士で納得し合いつつも，何だかいつも行楽地で会議している気がしないこともないですが…。

大倉山といえば，大倉山ジャンプ競技場があり，第11回冬季オリンピック札幌大会のジャンプ競技が開催された場所です。年に数回サマージャンプや，冬季ジャンプが開催されている全国的にもわりと有名な場所ですね。

車に揺られること20分，夏休みということもあって，混み合う駐車場になんとか到着。駐車場の近くにある長いエスカレータに乗ってジャンプ台の着地地点(観客が手旗を振って応援するあたり）に着いた感想は，「でっ，でかい…」。そして，きれいに整備されているのに驚きました。ここで舟木や原田が飛んだんだな～と思うとちょっと感動です。こんなところが札幌にあったなんて，とまた少し感動。観光地なので，聞いたことのない言葉が聞こえてきたりして，外国からの観光客も見かけました。

ジャンプ台の頂上には展望台があるのですが，そこへはリフトに乗って移動します。思わず気分は「クターのリフト」。リフトに乗るのは久しぶりなので，リフトが近づくと「う～，来た来た～っ！」と心の中でちょっぴりドキドキします。乗るときに足を前に出さないと膝をヤラレるので，初心者の人は注意してください。

展望台からは札幌市内の景色が一望できます。素晴らしく眺めが良いですよ。「あれ，札幌ドームだ！でかいUFOみたい」「あのへんは植物園？」「あそこは北海道大学か」「こんなところに大通公園あるの？意外と近い？」「すぐそこに見えるのって円山球場？」など，メンバー一同ウキウキしっぱなしで，地元に住んでいるにもかかわらず，ちょっとした観光旅行気分を満喫してしまいした。

いつもならデジカメを持ってきて今後のゲームの参考にするところなんですが，うっかり忘れてきてしまったのが非常に悔しいです。

帰りはリフト代をちょっとケチって歩いて山道を下ったのですが，日頃の運動不足のせいか，約1名膝がガクガクになっていました。大自然の中を散策できたので，精神衛生上は良かったみたいでした。

歩き疲れてジャンプ台下のレストハウスの休憩コーナーでしばらくまったりとしていましたが，重大なことに気がつきました。そうです，まだ会議が済んでなかったんです（いつも大体会議で遠出するときは会議のことは忘れています)。

それから約2時間くらい，男だけのメンバーで真剣な話し合いをしていたので，他の観光客から変な目で見られていないかちょっと不安でしたね。

会議が一段落着きそうになったので，レストハウスの向かいに建っている記念館を見物しに行きました。ここには大倉山ジャンプ台をリアルに体験できる『ジャンプシミュレータ』があると聞いていたので，メンバー一同飛ぶ気マンマンで入館したんですが，閉館時間の都合で乗れませんでした。トホホ…。

それでも，いろいろな展示物を見たり，ウインタースポーツの体験コーナーを片っ端から遊び倒し，極め付けに4人乗りボブスレーに全員で乗って，周囲の失笑を買いながらも，クールランニングな気分を堪能したのでした。

この先，制作するゲームに今日の体験記が生かせればいいなぁとか思いつつ，そんなことはお構いなしに遊びまくったギガ連射一行でしたとさ。

『ありがとう１周年！』

早いもので，今月号で『ギガ連射　つつうらら』も連載1周年を迎えました。これもひとえにみなさんがギガ連射を応援してくださったからですね。本当にありがとうございます。

今までギガ連射の制作裏話などを中心につらつらと書き連ねてきましたが，みなさんには「こんな感じでクターのゲームが作られてるんだなぁ」と少しでも雰囲気を感じてもらえれば本望です。

感想･質問など，みなさんからの熱～いメールお待ちしています。
otayori@giga-rensya.com
ギガ連射WEBサイト
http://www.giga-rensya.com/

※さてさて，じつは今回で『つつうらら』は最終回なんですが，次々号から心機一転して，新しい企画で進めたいと思います。それじゃまた12月号でお会いしましょう！！

Falcom® NEWS EXPRESS

日本ファルコム株式会社
〒190-0012 東京都立川市曙町1-14-13
TEL 042-527-6501
FAX 042-528-2714
スタッフ募集中
採用担当BFN係
http://www.falcom.co.jp/

FMS第4弾，9月13日発売！予約受付中！



新譜 ■交響曲ガガーブトリロジー
〜白き魔女〜朱紅い雫〜海の檻歌〜



■イースI・II
〜米光亮全曲集
パーフェクトコレクション (2枚組)



■イースI・II
〜スーパーアレンジ全曲集
パーフェクトコレクション (2枚組)

●試聴版を無料ダウンロードサービス実施中！

ファルコムホームページのCDコーナーでは，ただ今FMS第4弾の各アルバムのおいしいところを選りすぐった試聴版をダウンロードできます。ぜひ聴いてみてね。
http://www.falcom.co.jp/cd/
http://www.falcom.com/cd/

●下記サイトでも試聴版をダウンロードできます。

ベクター	: http://www.vector.co.jp/vpack/filearca/win/art/music/game/index.html
GAMEWATCH	: http://www.watchi.impress.co.jp/game/docs/20010803/demo0803.htm
cg	: http://eg.nttpub.co.jp/dl/index.html
4Gamer.net	: http:/www.4gamer.net/patch/files/main.html#sounds

PCゲーム定番キャンペーン！9月下旬よりスタート！

キャンペーン期間中にファルコムのゲームソフトを購入された方に，モバイル対応最新型マウスシートをプレゼント！厚さわずか0.5㎜なので段差がなく，裏面が特殊粘着加工のためデスクやテーブルなど平らなところならどこにでもピッタリと張りつきますので，デスクスペースを有効に使えます。また軽く，薄いのでノートパソコンの蓋に挟めて持ち運びも簡単！表面は特殊クリヤー仕上げのため耐摩擦性に強く，軽快なマウスタッチを保証！



ファルコムスタッフも
愛用
実用性保証済！

マウスシートプレゼント対象商品

- ■イースI・II完全版
- ■イースIIエターナル
- ■新英雄伝説III白き魔女
- ■英雄伝説IV朱紅い雫
- ■英雄伝説V海の檻歌
- ■ソーサリアンオリジナル
- ■西風の狂詩曲
- ■ファルコムゲームBOX　Vol.1

＊デザインは変更される場合があります。



ベクタープロレジにソーサリアンForever登場！

ベクタープロレジに，またまた新たなラインナップ！昨年「ソーサリアンオリジナル」が発売され大ヒットとなりましたが，「ソーサリアンForever」は，いわばソーサリアンの入門用として発売されたタイトル。ベクタープロレジ版には，ソーサリアンMIDIコレクションが特典として付いていますので，ソーサリアンをプレイしたことがない方は，まず「Forever」でゲームの流れを掴んで，「オリジナル」にチャレンジしてみては
http://www.vector.co.jp/



今月下旬からPCゲーム定番キャンペーンがスタートします。近日中にキャンペーン実施店をファルコムホームページのショップリストに掲載する予定ですので，近くに住んでいる方はチェックしておこう！

SEGAのヘンな話題やちょっとお得な情報ならおまかせ!!

ベーセガ

マイコンBASIC
SEGA OFFICIAL
VARIETY SPACE

奈美　みなさんこんにちは～。セガの竹崎＆ゲーマー奈美でーす。なんと，今月は羽田のセガの本社より中継でお送りしてますー。

竹崎　アー，アー，テス・テス・マイクテスト！

奈美　竹崎さん，なにやってんですか!?

竹崎　だって「中継」っていうからマイク……。

奈美　（ドカバキッ！）

竹崎　ふやぁ～。れも，りょっぴょんぎにいっこしたときよりは，なんか，りゃくらったれぇ，にゃみさん……。いてて……。

奈美　なんか荷物の量も少なかったですしね！

竹崎　通勤時間が一気に短縮した奈美さんは，これで毎日ばりばり残業モード突入～！

奈美　つまり，おたよりの宛先が変更になってますので，みなさん注意が必要ですーっと。

竹崎　おたよりの採用者には，特製の非売品クオカードを差し上げています！　どんどんおたよりくださいねッ！

奈美　…さて。PRも無事終了しましたところで，恒例のヤツ，行きますよ～。

竹崎　全国の大神さんに敬礼!?

奈美　（★☆★☆★☆!!!）

竹崎　ぐふぅぅぅ～～～っっっ!!（バタ！）

奈美　竹崎さーーん!! たーいへんですぅ!!

竹崎　こっ，恒例のってこのことかぁ……。

奈美　いよいよ「シェンムーII」が発売になりましたっ！

竹崎　発表から今日まで，ホントに長かった！

奈美　ですよねっ！　一体どういう展開になっているのかなぁー。

竹崎　そうそう，涼が香港の朋友（ポンユウ）に会いに行くんだよね！

奈美　あう。刃武鷹(レン・ウーイン)は朋友じゃありませんよ！　ぷんぷん。それと今作ではヒロイン，玲莎花（レイ・シェンファ）も登場してるんですよー。

竹崎　「シェンムーII初回限定版」にはもれなく「バーチャファイター4」特製プレミアムディスク（2枚組）も同梱されているぞ。この機会を見逃すなッ！

奈美　特製プレミアムディスクの内容ですが，ネットに接続して専用コンテンツを楽しめる「Virtua Fighter 4 Passport VF.NET」と。

竹崎　バーチャシリーズの並びに「VF1 プロトタイプ」の映像などが収録された「Virtua Fighter History / VF4」の二つだ。

奈美　太っ腹ですね！　そしてそして～。竹崎さん，とうとう出ますよ！　アレが！

竹崎　おうよ！

奈美　買ったらやらせてくださいよー。

竹崎　もちろんさ。しかし，小さいよねー。

奈美　持ち運びにも便利ですよね。

竹崎　懐かしいよね～。時代はめぐるというか，今時はこういうのが流行るのかな？

奈美　へ？　懐かしい？？？

竹崎　クルクル回して6面の色を揃えるだけなんて，単純かつ，奥が深いッ！

奈美　…それは？

竹崎　ルービック・キューーーブ！

奈美　（ドカべきバシぃ～ッ！！）

竹崎　……ママ，僕はママに愛されたいだけなんだ……。青い妖精はどこ？

奈美　しっかりしてくださいッ！！

竹崎　!?　殴ったのはあんたじゃッ！

奈美　失礼しました～。

竹崎　なげーよ！　フリが！！（笑）

奈美　ハイ，9月14日に任天堂さんからゲームキューブが発売になるわけですが，セガからは，バナナだ！　元気だ！　エイエイプ～！ゴールを目指しておサルを転がすアクションゲーム「スーパーモンキーボール」が本体と同時発売になりまっす。

竹崎　とにかく簡単操作ですぐ楽しめるぞ！

奈美　本編の他に，レースやボーリング，ビリヤード，ゴルフなどパーティゲームも豊富に収録されていますから，4人対戦プレイにはうってつけですよっ。

竹崎　ワイワイ皆で遊んでいるうちについ我を忘れてしまうぜ。

奈美　うっ。そういえば，先日撮影のために社内できゃーきゃー言いながら，4人対戦してたら，隣の部署からうるさい，というクレームが入ってしまいました（汗）。

竹崎　あらま。「気を付けよう，暗い夜道とモンキーボールの4人対戦」。

奈美　……。おあとがよろしいようで。

竹崎　よくね～よッ！　じゃ，また来月ぅ！



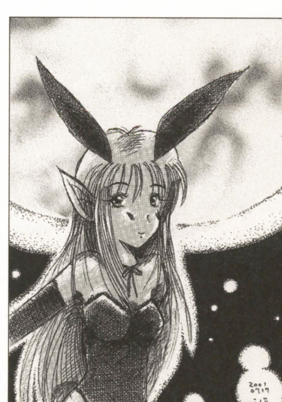
◀Aす

おひさです～。大人気のこみパですね。シェンムー2もよろしくです！



◀ARAI様江

こちらの気持ちを代弁していただいたようなイラストですね(笑)。おたよりお待ちしておりまーす



◀猿　ともこ

ご本のほう、送っていただきまして、ありがとうございました。PSOでお会いできるといいですね！



◀バグニンゲン

ふふん。…ご要望通り鼻で笑ってみました（汗）。ソニックはもちろん服の下に。違うですか？

このコーナーを充実させるための，ご意見，お便り，イラストなどなどクビを長くしてお待ちしています。採用の方には特製テレカなど，非売品セガグッズをプレゼント。SEGA竹崎がお便りをすべて読んで，面白いコーナーを作っていきたいと思います。すべてのあてさきは，ここや!!

〒144-8531　東京都大田区羽田1-2-12　(株)セガ　カスタマーゲートウェイ部「竹崎と奈美のベーセガ」係

セガ竹崎の「セガスタ」はこちら→http://www.sega.co.jp/segasuta/



タイトル冠文：大阪府・大阪難波
タイトル・イラスト：宮城県・吏久
本文：暁

全国256店舗の なにわ通信

PRESENTED BY マイコンソフト ADDRESS● 〒530-0005大阪市北区中之島3-2-4
朝日新聞ビル6F 電波新聞社大阪本社内
マイコンソフト株式会社　なにわ通信係
http://www.micomsoft.co.jp/

まだまだ暑いですが，皆様いかがお過ごしでしょうか？

先日，寝ころんだ状態で伸びをしたところ，「ゴリ…」と変な音がして，肋骨が折れてしまいました。体勢が悪かったのだろうと思います。皆さんも骨折には気を付けてください。

新製品の名前は？

諸般の事情で長らく名前を発表できなかった新製品ですが，この度「XRGB-2plus」という名称で改めてご紹介させていただきます。

XRGB-2plusは，パソコン用のディスプレイに家庭用ゲーム機などの映像を映す変換装置です。

パソコン用のディスプレイをゲーム・モニターとして活用できるので，とても便利です。XRGB-2plusは水平同期周波数31kHz(640×480ドット)や水平同期周波数47kHz(1024×768ドット)の画面モードを持つディスプレイに接続することができます。

マイコンソフトでは，現在XRGB-2という製品を販売しておりますが，XRGB-2plusは，その後継機となります。

XRGB-2plusには，ビデオ端子，S端子，D1端子，コンポーネント端子(Y,PB,PR)，RGB端子(マルチ21ピン)を持つゲーム機やDVDプレイヤーを接続することができます。

XRGB-2とXRGB-2plusの仕様比較表をご用意しましたので，購入の際の検討材料にしていただければと思います。

発売予定は11月初旬です。価格は未定ですが，30,000円前後になると思われます。

ミニ・オフ会

先日，名古屋で小さなオフ会が行われましたので，ご報告させていただきます。

参加者は投稿者の猿ともこさん，時期オクレさん，のうぎょうアッパーさん，きのえくまさん他，計6名のささやかなものでした。

ミニオフ会合作イラスト

湯けむりカラオケ大会（約6名）@名古屋



XRGB-2とXRGB-2plusの仕様比較

機能		XRGB-2（現行品）	XRGB-2plus（新製品）
水平同期周波数15kHz 出力対応		○	×
水平同期周波数31kHz 出力対応		○	○
水平同期周波数47kHz 出力対応		×	○
入力端子	ビデオ（コンポジット）	○※1	○
	S	○※1	○
	D1（Y,Pb,Pr）	×	○
	RGB（マルチ21ピン）	○	○
	PC入力（スルー出力用）	○	○
出力端子	15ピン HD-SUB（15ピン3段・VGA仕様）	○	○
	15ピン D-SUB（15ピン2段・NEC PC-9801仕様）	○	×
映像変換処理		RGB方式24bit	RGB方式24bit
ガンマ補正機能		アナログ方式 3段階	デジタル方式 96段階
R・G・B個別ガンマ調整		×	○
RGB端子からのY,Pb,Pr入力対応（PS2対応）		×	○
シャープネス調整	ビデオ（コンポジット）	○	○
	S	○	○
	D1（Y,Pb,Pr）		○
	RGB（マルチ21ピン）	×	○
業務用ゲーム基板への対応		△	×
ビデオテープ再生映像への対応		△	△
リモコン対応		×	○

○=対応可能，×=対応不可，△=動作保証なし
※1　音声はビデオとS共有。

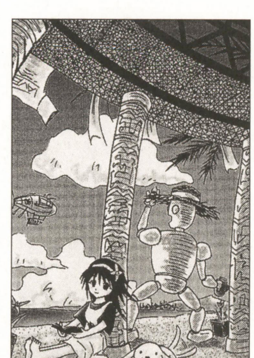
ありがたいおハガキのコーナー



愛知県・神奈月
はじめまして。ベートーベンの月光を聴いて，描かれた作品とのこと。もしや鳥人間でしょうか？



千葉県・バグニンゲン
「負けるな!なにわ課長シリーズ」の第1話です(笑)。麦藁帽子をかぶっている怪しい妖精が気になります。胸のXRGBロゴもGood！



大阪府・KARASU様江
爽やかですね～。こんな浜辺でゆっくり余生を過ごしたいものですね…。おっと，またオヤジくさいセリフが出てしまった。



大阪府・ARAI
躍動感に満ちたイラストです。沢山の想い出と共に彼は深い海の底へ…。衝撃的な事件が起きてしまったようです。

中古通販 PC NET ピーシーネット

在庫保有数日本一
(常時10,000台)

TEL.03-3733-1441

通販営業時間／月～金 9:30～19:00 土・日・祝日 11:00～19:30

www.prins.co.jp/

楽天市場店 http://www.rakuten.co.jp/pcnet/
ネットプライスモール http://www.netpricemall.com/
ジャストネット http://emall.justnet.ne.jp/shop/pcnet/
ビッグローブショッピングワールド http://shopping.biglobe.ne.jp/

最新情報は電話かネットでチェック!! 急げ!!

祭 IBMまつり 開催

●IBM商品が何でも大特価で大放出!!
●ネットまたはTELで要チェック!!

期間 9月1日(土)～9月30日(日)

ノートもディスプレイもデスクトップもビックリ価格!!

インターネットでは毎回お得な企画がいっぱい!!

3日間保証セール だから安い!!

期間限定タイムセール

[安心の全品保証付] 1万円以上…1ヵ月保証 1万円以下…1週間保証

※本商品は添付品（OSを含む）は付属しておりません。
※クリーニングご希望の方、別途¥1,000が必要です。
※中古パソコン販売時の在庫定義が変わりました。

今後在庫とは、従来品に加え、動作上支障のないキズ・天板やけ・カバー等の欠品がある物や、キーボード等が補充（他メーカー含）されている商品も、「在庫」としてお取り扱いさせていただきます。

DOS/V

注文No.	品名	価格
U2979	Celebris FX-2 5200M ST(DEC)(MP(200)/32MB/3.2GHD/CD16)	¥9,800
SH744	FR-G2B2J-EB(DEC)(PenII(233)/64MB/3GHD/CD24/Win95/付属品付)◆優良商品	¥26,800
SH742	FR-2B2J-EB(PC5510/6233M)(DEC)(PenII(233)/96MB/3.2GHD/CD/Win95/付属品付)◆優良商品	¥29,800
U8342	Aptiva(2190-22L)(IBM)(K6-2(450)/64MB/6.4GHD/CD40/省スペース型/PCI×3/USB×2)	¥31,800
U2501	PC300PL(6562-38J)(IBM)(MP(200)/32MB/4.2GHD/CD24)	¥9,800
U8612	6272-G3J(IBM)(MP(233)/32MB/4.2GHD/CD16/共有×2/USB×2)	¥18,800
U8615	6561-55J(IBM)(PenII(300)/32MB/4.2GHD/CD32/PCI×2/ISA×1/共有×1/USB×1)	¥24,800
SH748	PC300PL(6892-44J)(IBM)(PenII(350)/32MB/6.4GHD/CD32/Win95/ミドルタワー型/付属品付)◆優良商品	¥36,800
U8618	PC-MA26DSHAAB42(NEC)(PenII(266)/64MB/4GHD/CD/外装塗装剥げ)	¥17,800
U8630	PC-MA33D/M-42(NEC)(PenII(333)/64MB/4.3GHD/CD24/外装塗装剥げ/ミドルタワー型/PCI×5/USB×2)	¥20,000
U8646	PC-MA36HC(NEC)(Celeron(366)/160MB/4.3GHD/CD24/省スペース型/PCI×1/USB×2)	¥26,800
U8519	PC-MA43H/FL(液晶一体型)(NEC)(Celeron(433)/64MB/6.4GHD/CD24/15TFT(1280×1024)/LAN/外装焼け/一体型/USB×2)	¥79,800
U8598	PC-MA23D/C(NEC)(PenII(233)/32MB/3.2GHD/CD20/省スペース型/PCI×3/USB×2)	¥17,800
U1193	PC-MA33D/C(NEC)(PenII(333)/64MB/4.3GHD/CD20/LAN/省スペース型/PCI×2/USB×2)	¥26,800
SH486	DESKPRO4000S/5233X(COMPAQ)(MP(233)/32MB/2.1GHD/CD20)	¥17,000
SH719	DESKPRO EN SF6350(COMPAQ)(PenII(350)/64MB/6.4GHD/CD24/NT4.0/付属品付)◆優良商品	¥28,000
U8659	DESKPRO DPENS P500/10/64/W5C(COMPAQ)(PenIII(500)/128MB/10GHD/CD/省スペース型)	¥40,800
U1709	OPTIPLEX GXI(DELL)(MP(200)/32MB/2.1GHD/CD8/擦り傷/高さ10.9cm/PCI×1/USB×2)	¥8,800
U8426	OPTIPLEX GXI(DELL)(MP(200)/32MB/2.1GHD/CD8/外装焼け/汚れ/高さ16.5cm/USB×2)	¥18,000
U1789	OPTIPLEX GX1(DELL)(PenII(400)/32MB/3.2GHD/CD/外装焼け/擦り傷/タワー型/USB×1)	¥36,800
U8664	OPTIPLEX GX1(DELL)(PenII(400)/64MB/4.5GHD/CD/MO付(保証対象外)/高さ17cm)	¥29,800
U1865	Dimension XPS D266(DELL)(PenII(266)/32MB/3.2GHD/CD/タワー型/PCI×4/ISA×2/USB×2)	¥18,800
U8665	GATEWAY2000(G6-266)(GATEWAY)(PenII(266)/64MB/3.2GHD/CD/外装塗装剥げ/汚れ/高さ48cm)	¥19,800
U1245	VectraVL6/333-7(HP)(PenII(333)/80MB/4.3GHD/CD24/外装焼け/外装キズ/PCI×1/USB×2)	¥20,800
V0144	ZHP-D6582N/801(HP)(PenII(400)/128MB/6.4GHD/CD32/100BT)	¥33,800
SH767	FMV-S52373(富士通)(PenII(233)/32MB/3.2GHD/CD24/Win95/付属品付)◆優良商品	¥27,800
U1923	FMVT53071.2.3(TV307)(富士通)(PenII(300)/96MB/4.3GHD/CD32/スピーカ無し/モデム/外装焼け・キズ/MO付(保証無)/USBx2)	¥26,300
U8603	PC-5DC02-PFOMA,B(日立)(MP(233)/64MB/4.3GHD/CD20/LAN/省スペース型/PCI×1/共有×1/USB×2)	¥19,800
U8609	PC1DC6-A,E,B(6012)HC,HAOO(日立)(Celeron(366)/32MB/4.3GHD/CD24/立置きスタンド/省スペース型/PCI×2)	¥29,800
U1748	PROMATE P3233ML/95(PB-NEC)(PenII(233)/32MB/3GHD/CD20/外装焼け/KB無し/タワー型/USB×2)	¥11,800
U8605	PORTLAND PII266-MT(MICRON)(PenII(266)/64MB/6.4GHD/CD/ミドルタワー型(43cm)/USB×2)	¥24,800
U8631	MicroPCSTATION M250A(SOTEC)(Celeron(500)/128MB/10GHD/DVD/スピーカ無し/PCI×3/USB×2)	¥41,800

DOS/Vノート

注文No.	品名	価格
U6810	ThinkPad365X(2625-BJ9)(IBM)(Pen(150)/40MB/1.4GHD/SVGA/KB文字剥げ/11.3TFT)	¥29,800
U3313	ThinkPad755CX(9545-SFJ)(IBM)(Pen(75)/8MB/540HD/VGA/擦り傷/PCカバー無し/10.4TFT)	¥8,900
U6802	ThinkPad770(9549-1AJ)(IBM)(MP(233)/32MB/5.1GHD/XGA/外装塗装剥げ/14.1TFT/USB×1)	¥55,800
A0085	PC-VP16CWSAA1(NEC)(MP(166)/32MB/2.1GHD/CD20/SVGA/12.1TFT)	¥41,000
U6670	PC-VA20C/BS(NEC)(MP(200)/64MB/3GHD/モデム/SVGA/10.4TFT/USB×1)	¥49,800
U6549	PC-VA26H/WSAA(NEC)(Celeron(266)/64MB/4.3GHD/CD無し/外装キズ/外装塗装剥げ/SVGA/12.1TFT)	¥57,800
U6821	ARMADA1510(COMPAQ)(Pen(120)/32MB/1.GHD/SVGA/外装擦り傷/PCカバー無し/11.3DSTN)	¥10,000
V0156	ARMADA1500C(388066-294)◆Win98(COMPAQ)(Celeron(366)/64MB/4.0GHD/CD24/SVGA/12.1TFT/付属品付)	¥78,800
U6596	LATITUDE XPI P133ST(DELL)(Pen(133)/40MB/1.4GHD/VGA/11.3TFT)	¥17,800
U3237	FMV-450N/S1E3(富士通)(486DX2(50)/3.6MB/340HD/PCカード欠/外装焼け/擦り傷/DSTN)	¥6,000
U3472	FMV-47NT2E(富士通)(486DX4(75)/12MB/340HD/外装キズ・PCカバー欠品/10.3TFT)	¥6,800
U6336	FMV-57NUYH(FMV-575NU/Y)(富士通)(Pen(75)/24MB/810HD/SVGA/A4/外装キズ/10.4DSTN/16MB増設)	¥6,800
U6274	FMV-5120NU2/W(FMV-52NU2W5.1共通)(富士通)(Pen(120)/32MB/1GHD/SVGA/外装キズ/外装焼け/11.3TFT)	¥14,800
U3463	FMV-53NU6W1,5(富士通)(Pen(133)/32MB/1.3GHD/SVGA/11.3TFT)	¥18,800
U3941	FMV-53NU5S◆お買い得商品(富士通)(Pen(133)/32MB/1GHD/CD10/ﾓﾃﾞﾑ/SVGA/擦り傷/11.2TFT)	¥27,800
U3483	FMV-53NP6W5(富士通)(MP(133)/32MB/1.6GHD/CD16/SVGA/12.1TFT)	¥41,800
U3610	FMV-55NA5W5(富士通)(MP(150)/32MB/1.6GHD/CD10/SVGA/外装キズ/12.1TFT)	¥43,800
U6487	FMV-5BNA6X7,5(FMV-5233NA6/X)(富士通)(MP(233)/32MB/3GHD/CD20/XGA/擦り傷/焼き焦げ/13.3TFT/USB×2)	¥49,800
U6801	DynaBook PA1261CN(東芝)(MP(166)/32MB/2.1GHD/SVGA/12.1TFT)	¥38,000
U3046	DynaBook PAP300JA(東芝)(MP(233)/64MB/2.1GHD/SVGA/PCカバー欠/12.1TFT/B5サイズ)	¥59,800
U6517	DynaBook PA1270C9(東芝)(PenII(266)/64MB/4GHD/CD20/XGA/外装塗装剥げ/13.3TFT/USB×1)	¥57,800
U6688	DynaBook PAP333J8,W(東芝)(Celeron(333)/64MB/4.3GHD/XGA/外装塗装剥げ/PCカバー無し/11.3TFT/USB×1)	¥70,800
U6648	DynaBook Satellite2210(PA-SA50C4C8)(東芝)(Celeron(500)/64MB/6GHD/CD24/XGA/14.1TFT/USB×1)	¥108,000
V0122	PC-5NH02-PD5LA(日立)(MP(233)/32MB/3.2GHD/24CD/100TX/12.1TFT)	¥55,000
V0188	PC1NH5-E4D22R320(FLORA270EX)◆Win98付(日立)(PenII(333)/64MB/4.3GHD/CD20/LAN/13.3TFT/付属品付)	¥92,800
V0194	PC1NH8-G7D22H110(FLORA270SX)◆Win98付(日立)(Celeron(400)/64MB/4.8GHD/CD24/100TX/13.3TFT/付属品付)	¥105,800
U3830	AL-N1T515J5(パナソニック)(Pen(150)/16MB/1.35GHD/SVGA/PCダミーカバー無し/外装塗装剥げ/10.4TFT)	¥39,800
U6124	AL-N2T520J5(パナソニック)(MP(200)/32MB/2.1GHD/SVGA/擦り傷/外装塗装剥げ/10.4TFT)	¥49,800
U6682	MEBIUS PC-PJI(シャープ)(MP(233)/64MB/3.2GHD/SVGA/擦り傷/PCカバー無し/11.3TFT/USB×1)	¥69,800
U6396	SINVA350(キヤノン)(MP(166)/32MB/1.44GHD/CD8無し/モデム/XGA/13.3TFT/CD着脱式)	¥32,800

バッテリーパックは消耗品につき保証対象外とさせていただきます。

98デスクトップ

注文No.	品名	価格
U2627	PC-9821V200/SZC2(MP(200)/32MB/3GHD/CD20/スピーカ無し/外装焼け/外装塗装剥げ)	¥28,800
C0088	PC-9821V200/S7D2(MP(200)/32MB/2GHD/CD20/33.6K)	¥39,800
U2632	PC-9821V233/M2(MP(233)/32MB/3GB/CD20/Fax36.6K/ミニタワー/外装焼け・キズ/スピーカ・マイク無し)	¥42,800
U2760	PC-9821CR13/TA(Pen(133)/16MB/850HD/CD4/14.4Kモデム/TV/TFTカラー液晶/擦り傷/一体型)	¥34,800
U8347	PC-9821XA20/W30(Pen(200)/64MB/3GHD/CD8/PCM/PCI×2)	¥19,800
U8350	PC-9821XA20/W30(Pen(200)/96MB/3GHD/CD8/PCM/64MB増設/外装焼け/外装キズ/PCI×2)	¥23,800
SH620	PC-9821RA20/N30(PPro(200)/32MB/3GHD/CD8/WinNT/外装キズ/付属品付)◆優良商品	¥24,800
U8349	PC-9821RA223/W30R(PenII(233)/64MB/3GHD/CD16/天板外装キズ/外装焼け/PCI×2)	¥25,800
U1584	PC-9821RA266/W30(PenII(266)/32MB/3GHD/CD16/キーボード文字剥げ)	¥36,800
U8354	PC-9821RA266/W・D30R(PenII(266)/64MB/3GHD/CD16/32MB増設/PCI×2/ISA×3)	¥38,800

98ノート

注文No.	品名	価格
U3059	PC-9821ND(486SX(33)/1.6MB/外装焼け/9.5DSTN)	¥12,800
U3420	PC-9821ND(8/340)(486SX(33)/8MB/340HD/9.5DSTN/メモリ・HD増設)	¥12,800
U3334	PC-9821NF/810(Pen(75)/7.6MB/810HD/PCM/9.5TFT)	¥13,800
SH754	PC-9821NS/540W(486DX2(50)/5.6MB/540HD/Win3.1/9.5TFT/付属品付)◆優良商品	¥19,800
U6278	PC-9821NP/540(486DX4(75)/38MB/540HD/VGA/擦り傷/9.5TFT/32MB増設)	¥14,800
U3027	PC-9821LA10/8A(Pen(100)/16MB/810HD/サブノート/PCダミカバー欠品/9.5TFT)	¥26,800
U3787	PC-9821LS150/S14(MP(150)/32MB/1.4GHD/CD10/VGA/擦り傷/PCカバー無し/12.1TFT)	¥45,800
U6321	PC-9821NB7/5(Pen(75)/24MB/540HD/VGA/PCM/外装焼け・キズ/10.4TFT)	¥11,000
U3323	PC-9821NA7/H7(Pen(75)/8MB/720HD/CD4/PCM/10.4TFT)	¥12,800
U3217	PC-9821NA9/H8(Pen(90)/16MB/810HD/CD4/PCM/外装キズ/KB文字焼け/10.4TFT)	¥16,800
U3386	PC-9821NA12/H10(Pen(120)/16MB/1GHD/PCM/SVGA/CD4/10.4TFT)	¥27,800
U3075	PC-9821NA13/H10(Pen(133)/16MB/1GHD/CD4/PCM/SVGA/12.1TFT)	¥31,800
U3275	PC-9821NR12/D10(Pen(120)/16MB/1GHD/外装擦り傷/外装塗装剥げ/12.1DSTN)	¥19,800
U6490	PC-9821NR13/S14(Pen(133)/32MB/1.4GHD/SVGA/擦り傷/12.1TFT)	¥27,800
U6331	PC-9821NR15/S10(Pen(150)/32MB/1GHD/CD8/SVGA/12.1TFT)	¥36,800
U6547	PC-9821NR15/S14F(Pen(150)/32MB/1.4GHD/CD10/SVGA/モデム/12.1TFT)	¥40,800
U3705	PC-9821NR150/S20(MP(150)/32MB/2.1GHD/擦り傷/12.1TFT)	¥34,800
U6052	PC-9821NR166/X30F◆お買い徳！(MP(166)/32MB/3GHD/CD10/XGA/13.3TFT)	¥49,800
U6165	PC-9821NR233S32A(MP(233)/32MB/3.2GHD/CD16/SVGA/外装キズ/外装塗装剥げ/12.1TFT)	¥64,800
U1647	PC-9821NW133/D14C.D共通(MP(133)/32MB/1.4GHD/CD11/SVGA/12.1DSTN)	¥24,800
U6075	PC-9821NW150/S20C(MP(150)/32MB/2GHD/CD16/SVGA/擦り傷/ドットカケ/12.1TFT)	¥38,800
U6097	PC-9821NW150/S20D(MP(150)/32MB/2GHD/CD16/SVGA/12.1TFT)	¥58,800
U6806	PC-9821NR233/S32A(MP(233)/32MB/3.2GHD/CD16/SVGA/外装擦り傷/12.1TFT)	¥68,800

バッテリーパックは消耗品につき保証対象外とさせていただきます。

MAC

注文No.	品名	価格
U9346	Performa 6310(本体のみ)(PPC603e(100)/16MB/1.2GHD/CD4/外装焼け)	¥8,000
U9413	PowerMac 6300/120(PPC603e(120)/32MB/1.2GHD/CD4/外装焼け・キズ)	¥9,800
U9090	PowerMac 7200/90(PPC601(90)/16MB/500HD/CD/外装焼け・キズ)	¥7,800
U9281	PowerMac 7200/120(PPC601(120)/32MB/1.2GHD/CD4/外装焼け・キズ)	¥10,800
U9351	PowerMac 7300/166(PPC604e(160)/16MB/2GHD/CD12/外装キズ・焼け(全体)/マウス無し)	¥12,800
U9344	PowerMac 7300/180(PPC604e(180)/16MB/2GHD/CD12/マウス無し/外装焼け)	¥21,000
U0287	PowerMac 7500/100(PPC601(100)/16MB/1GHD/CD/外装焼け/ファイルカバー無し)	¥7,800
U9190	PowerMac 7500/100(PPC601(100)/32MB/1GHD/CD/外装焼け)	¥12,800
U9396	PowerMac 7600/120(PPC604(120)/49MB/1.2GHD/CD4/外装焼け)	¥18,000
U9342	PowerMac 7600/200(PPC604e(200)/32MB/2GHD/CD12/外装焼け/擦り傷)	¥18,000
U9443	PowerMac 8100/100(PPC601(100)/32MB/700HD/CD2/外装焼け/外装キズ)	¥19,800
U2159	PowerMac 8100/100AV(PPC601(100)/16MB/1GHD/CD2/マウス欠品)	¥9,800
U2054	PowerMac 8500/120(PPC604(120)/32MB/2GHD/CD)	¥23,000
U9244	PowerMac 8500/120(PPC604(120)/80MB/2GHD/CD4)	¥25,000
U9175	PowerMac 8500/132(PPC604(132)/64MB/1.2GHD/CD)	¥26,000
U9431	PowerMac 8500/150(PPC604(150)/49MB/1.2GHD/CD8)	¥27,800
U2212	PowerMac 8500/180(PPC604e(180)/32MB/2GHD/CD8/外装焼け・キズ/マウス無し)	¥32,800
U9358	PowerMac 8600/200(PPC604e(200)/32MB/2GHD/CD/ZIP付(動作保証無し)/外装焼け)	¥46,800
U9387	PowerMac 9500/120(PPC604(120)/64MB/1GHD/CD4)	¥34,800
U9296	PowerMac 9500/132(PPC604(132)/32MB/2GHD/CD4)	¥36,800
U9301	PowerMac 9500/150(PPC604(150)/32MB/2GHD/CD4/外装焼け)	¥41,800
U9427	PowerMac 9600/233(PPC604e(233)/80MB/4GHD/CD12/外装焼け/メモリ増設)	¥51,800
U1367	PowerMac G3/DT266(PPCG3(266)/32MB/4GHD/CD24/ZIP付き/外装キズ)	¥49,800
U9345	PowerMac G3/MT266(PPCG3(266)/32MB/6GHD/CD24)	¥64,800
U9398	PowerMac G3-350(PPCG3(350)/64MB/6GHD/CD32)	¥99,800
U9476	PowerMac G4/400(PPCG4(400)/64MB/10GHD/DVD)	¥138,000
U9128	PowerMac G4/500(ｸﾞﾗﾌｧｲﾄｶﾗｰ)(PPCG4(500)/256MB/27GHD/DVD-RAM/ミドルタワー型/USB×2)	¥168,000
U8569	H6D0T(UMAX)(PPC604e(225)/160MB/2GHD/CD/本体カバー爪折れ/ミドルタワー型/PCI×6)	¥38,800

キーボードは別売となります。

PowerBook

注文No.	品名	価格
U3562	PowerBook 190C(68LC040(66)/8MB/500HD/10.4DSTN)	¥15,000
U3063	PowerBook 1400C(PPC603e(133)/16MB/1GHD/CD/11.3CTFTカラー)	¥41,800
U9466	PowerBook 1400CS/166(PPC603e(166)/48MB/1.3GHD/CD8/SVGA/11.3FTSN)	¥54,800
U6133	PowerBook 1400CS/117(PPC603e(117)/16MB/770HD/CD6/SVGA/11.3DSTNカラー)	¥39,800
U3066	PowerBook 5300C(PPC603e(100)/32MB/750HD/外装擦り傷/10.4CTFTカラー/メモリ増設)	¥28,000

ディスプレイ

注文No.	品名	価格
SH779	F15R1W(NEC)(15TFT：1024×768max/3列15pin)	¥42,800
SH444	PC-KM173R2(NEC)(17マルチ：1280×1024max/0.28mm/98MS-DOS使用可能/3列15pin)	¥12,800
U4505	DV17C4(NEC)(17マルチ：1280×1024max/外装焼け/3列15pin)	¥12,800
U4830	TFT5000S(COMPAQ)(15.1TFT：1024×768max/外装キズ/リアカバー無し/3列15pin)	¥38,800
U4576	M770(DELL)(17マルチ：1024×768max/外装キズ・焼け/3列15pin)	¥11,000
U4843	MA-2091(NANAO)(20マルチ：1600×1200max/汚れ/3列15pin)	¥24,800
U4829	FlexScan L66(NANAO)(18.1TFT:1280×1024max/ディスクトップタイプ)	¥98,800
U4845	CE-LT14M(シャープ)(13.8TFT：1024×768max/外装焼け/3列15pin)	¥28,800
SH710	CPD-17C1(SONY)(17マルチ：1280×1024max/外装焼け・キズ/3列15pin)	¥12,800
SH446	CPD-17GS(トリニトロン)(SONY)(17マルチ：1280×1024max/0.25mm/外装キズ/外装焼け/3列15pin)	¥11,800
U4697	GDM-19PS(SONY)(19マルチ：1280×1024max/3列15pin)	¥24,800
U4844	CPD-G400J(SONY)(19マルチ：1800×1440max/外装キズ/外装焼け/3列15pin)	¥26,800
U4590	GDM-20SE1(SONY)(20マルチ：1280×1024max/0.30mm/5BNC)	¥30,000
D0132	FMVCRT(17)(富士通)(17マルチ：1024×768max/3列15pin)	¥9,800
U4799	RDT-141R(三菱)(14TFT：1024×768max/約26万色/3列15pin)	¥38,800
U4529	RD-17M(三菱)(17マルチ：1280×1024max/0.28mm/3列15pin)	¥14,000
U4519	RD-21GII(ダイヤモンド)(三菱)(21マルチ：1600×1200max/0.28mm/3列15pin/5BNC)	¥30,000
SH768	A701G(飯山電機)(17マルチ：1600×1200max/3列15pin/5BNC)	¥13,800
U4369	MT-8617ES(飯山電機)(17マルチ：1600×1200max/0.25mm/3列15pin/5BNC)	¥12,800
U4567	MT-8521E(飯山電機)(21マルチ：1600×1200max/0.30mm/3列15pin/5BNC)	¥29,800
U4834	TXA3602GT(飯山電機)(14.1TFT：1024×768max/外装焼け/3列15pin)	¥35,800
U4827	PC-DT3150(日立)(15.5スーパークリアカラー：1024×768/外装焼け/3列15pin)	¥45,800
U4777	CF-L15TCJM(パナソニック)(15TFT：1024×768max/3列15pin)	¥40,000
U4737	GATEWAY 500CS(GATEWAY)(15マルチ：1024×768max/3列15pin)	¥6,800

プリンタ

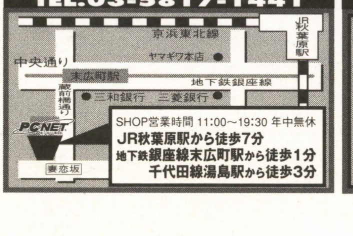
注文No.	品名	価格
SH543	PR-L2000X2(NEC)(A4/1500DPI:マルチライタープリンター/付属品付)	¥12,400
SH771	LP-9000PS2F2(エプソン)(レーザー：A3/PS対応/付属品付)	¥39,800
P0142	LBP-750(キヤノン)(レーザー：A3対応)・・・・・・・・	¥25,800
U5152	LBP-2030(キヤノン)(レーザー)・・・・・・・・・・・・・	¥48,000

トナー等は消耗品につき標準添付されておりません。別途、ご確認下さい。

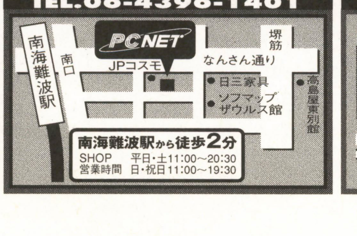
商品の状態により価格の変動がある事がございます。
詳しくはTELかネット上でご確認ください。
万一、品切れの際はご容赦ください。(8月28日現在)

こちらの在庫リストは通販専用となります。店舗在庫は各店舗へご来店の上ご確認ください。

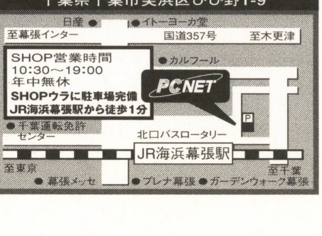
秋葉原店
千代田区外神田6-5-12 偕楽ビル新末広2F(1Fジョナサン)
TEL.03-5817-1441



大阪日本橋店
浪速区難波中2-1-7 なんば東ビル
TEL.06-4396-1401



幕張店
千葉県千葉市美浜区ひび野1-9



パシフィック ネット
株式会社パシフィックネット
本社 〒144-0051 東京都大田区西蒲田7-10-1
相互西蒲田ビル TEL.03-3733-1441 FAX.03-3733-1440
通販営業時間／月～金 9:30～19:00 土・日・祝日 11:00～19:30

計測機器を各ショップのみで販売中。
詳しくは各ショップまでお問い合わせください。

iモードでショップ最新情報検索
http://www.prins.co.jp/i/

◆ 定期購読●直接購読のしおり ◆

弊社刊行の雑誌は，ご自宅直送方式の年間定期購読をお勧め申し上げます。スピーディーで確実，そしてお得な定期購読をぜひご利用下さいませ。

また ご希望の雑誌・書籍が，ご近所の書店にない場合は右の振替用紙にて直接ご注文下さい。

■ 年間定期購読料（消費税込料金となっております。）

電　気　店	￥16,000	マイコンBASICマガジン	￥6,120
電波新聞縮刷版（CD-ROM版）	￥96,000		

■ 年間購読の特典

①特別号も通常の定価でご購読いただけます。
②電気店は13冊分の購読ができます。

◆確実な振替口座を利用してご送金下さい◆

送金手数料が不要で，確実な郵便振替でご送金をお願い申しあげます。手続きは簡単です。右の郵便振替用紙に下記の必要事項を記入し，郵便局窓口に必要金額と共にご提出いただくだけですみます。

①購読申込書に必要事項を記入ください。なお1年間，或は2年間購読かをはっきりご記入ください。
②払込通知票，払込票に送金合計金額と住所，氏名をご記入ください。払込通知票と払込票は絶対に切りはなさないようにしてください。なお，必ずフリガナをご記入下さい。
③弊社への御意見・御希望がございましたら，通信欄へご記入下さい。
④すべて記入が終了しましたら，案内文のみ切り離して郵便局窓口に金額を添えて提出してください。

∴振替によるご送金は，着本までに約2週間ほどかかりますのでお含みおきください。
お急ぎの場合は，現金書留に，右購読申込書を同封してご送金ください。

この受領証は、郵便局で機械処理をした場合は郵便振替の払込みの証拠となるものですから大切に保存してください。

ご注意

この払込書は、機械で処理しますので、口座番号及び金額を記入する際は、枠内に丁寧に記入してください。

また、下部の欄（表面及び裏面）を汚したり、本票を折り曲げたりしないでください。

（郵政事業庁）

払込取扱票

02 東京

通常払込料金 加入者負担

口座番号（右詰めにご記入ください）	金額	千 百 十 万 千 百 十 円
00150-3-51961	料金	特殊取扱

加入者名：株式会社 電波新聞社

通信欄

※定期購読誌名	購読期間	送金額（送料当社負担）
※新規 継続	年 月号より 年間	円
御希望誌・ソフト・グッズ名	冊数	送金額（送料共）
		円
		円

フリガナ 送付先住所 □□□-□□□□	産業分類	所属又は職種
フリガナ 送付先御氏名	電話 TEL	年令 才	性別 男・女

払込人住所氏名（郵便番号 ）※

（電話番号 － － ）

受付局日附印

裏面の注意事項をお読みください。郵政事業庁（私製承認 東 第1005号）

各票の※印欄は、払込人において記載してください。

切り取らないで郵便局にお出しください。

記載事項を訂正した場合は、その箇所に訂正印を押してください。

払込票兼受領証

口座番号	00150-3 通常払込料金加入者負担
	右詰めにご記入ください 51961
加入者名	株式会社 電波新聞社
金額	千 百 十 万 千 百 十 円 ※
払込人住所氏名	※
料金	（消費税込み） 円
特殊取扱	

受付局日附印

切り取って郵便局へお出し下さい。

電波新聞社出版物一覧

（消費税込料金となっております）

2001.7 現在

書名	価格	〒	書名	価格	〒
最新エレクトロニクス用語事典	5,300円	〒380円	DOS/V制作スペシャルNo.2～11	1,490円	〒340円
半導体・ICのすべて	2,520円	〒310円	F-BASIC V6.3プログラミングのすべて	2,835円	〒310円
ビデオ技術のすべて	2,100円	〒310円	ネットワーク対応データベース構築ガイド	2,854円	〒310円
TVのすべて改訂版	1,988円	〒310円	パソコン認定試験合格対策試験問題集3・4級	2,100円	〒310円
CD技術のすべて	2,039円	〒310円	デジタル1種〈技術科目〉最新テキスト	2,205円	〒340円
MDのすべて	1,400円	〒310円	デジタル3種受験マニュアル	3,000円	〒340円
センサーのすべて	2,000円	〒310円	アナログ・デジタル全資格〈法規科目〉	1,995円	〒340円
改訂版DVD&DVC入門基本18章	1,529円	〒310円	シスアド受験スーパー記憶術	2,900円	〒340円
増補改訂版デジタル放送がわかる本	1,600円	〒310円	基本情報受験スーパー記憶術	2,900円	〒340円
増補改訂版・新ビデオ技術ハンドブック	2,500円	〒310円	特殊無線技士受験マニュアル・改訂新版	2,905円	〒340円
新・まんが電気学入門	1,500円	〒310円	デジタルビデオカメラを使いこなす本シリーズ ビクター・ソニー・パナソニック・シャープ・キャノン	各 1,490円	各 〒310円
新電気学入門講座	2,050円	〒310円			
新オシロ・ロジアナ入門講座	2,243円	〒310円	オールアバウト カプコン対戦格闘ゲーム1987～2000	2,100円	〒450円
電子計測器ガイドブック	7,500円	〒450円	オールアバウト SNK対戦格闘ゲーム 1991～2000	2,310円	〒450円
IT社会のビジネスプラットホーム	2,625円	〒310円	IT革命にどう生き残るか	2,205円	〒310円

※上記以外にも新聞・出版物がございますので最寄りの本社、支局へお問い合わせ下さい。

東京本社 03-3445-8201(ダイヤルイン)
大阪本社 06-6203-3361
西部本社 092-431-7411
札幌支局 011-641-5591
仙台支局 022-227-7211
新潟支局 025-245-2526
関東総局 027-326-3206
松本支局 0263-36-2266
名古屋支局 052-261-4541
金沢支局 076-263-8661
京都支局 075-221-8021
広島支局 082-228-5581
高松支局 087-861-3111
鹿児島支局 099-226-3630

Q10　携帯・PHSを持っていますか？
①両方持っている ②携帯 ③PHS ④購入予定 ⑤予定なし

Q11　Q10で④の方のみ，どちらを購入する予定ですか？
①携帯 ②PHS ③両方 ④未定

Q12　使用中のプログラミング言語は？
※枠内に一つご記入ください

Q13　これから勉強したいプログラミング言語は？
※枠内に一つご記入ください

Q14　今月号の本誌で面白かった記事を三つ，下の表からお答えください

Q15　今月号の本誌でつまらなかった記事を三つ，下の表からお答えください

ご協力ありがとうございました。

■今月号の記事一覧

マイコンBASIC Magazine　プログラム投稿応募カード

言語名（音源名）					
プログラム名（曲名）					
氏　名		ペンネーム		年齢	歳
住　所	〒				
TEL	※連絡可能な時間帯			応募年月日	年　月　日

キリトリ線

郵便はがき

141-8715

50円切手をお貼り下さい

（受取人）

東京都品川区東五反田1－11－15

電波新聞社

月刊 マイコンBASICマガジン　編集部

10月号 行

(フリガナ)		年齢	性別
ご氏名		歳	男・女
□□□-□□□□		電話	
ご住所	都・道 府・県		

職業：小学生／中学生／高校生／高専生／短大生／大学生／専門学校生／予備校生／アルバイト／公務員／会社員／その他（　　　　）

本誌を 毎月・時々・初めて 読む　｜　何人で読みますか（　　）人

自分のホームページを　持っている・持つ予定・持っていない・予定なし

本誌のほかによく読む雑誌は？［　　　　　］

ベーマガお楽しみプレゼント⑩ 応募欄（80ページ欄外）	希望番号	品名
	番	

キリトリ線

月刊マイコンBASICマガジン⑩ 愛読者カード

★OFコーナー宛 ♥あくせすROOM宛 ■質問・感想（本誌への掲載／可・不可）

ペンネーム：

■アンケート回答記入欄 ■ ※黒の筆記用具でわかりやすく，はっきりとご記入ください

Q1	□	Q2	□	〔メーカー名 〕				
Q3	□	Q4	□	Q5	□			
Q6	□	□	□	□	□	□	□	□
Q7	□	□	□	□	□	〔商品名 〕		
Q8	□	□	□	□	□	□	□	□
Q9	□	□	□	□	□	□	□	□
Q10	□	Q11	□					
Q12	〔 〕	Q13	〔 〕					
Q14	□	□	□	Q15	□	□	□	

キリトリ線

■アンケートにご協力ください。

※左の愛読者カード・記入欄に，以下の質問に該当する番号をご記入ください。複数回答の質問で，回答欄をすべて埋める必要はありません。

Q1　メインで使っているパソコンは？
①デスクトップ ②ノート ③持っていない

Q2　パソコンの種類は？（Q1で③以外の方）
①自作 ②ショップブランド ③メーカー製
※枠内にメーカー名も記入してください

Q3　メインで使っているOSは？（Q1で③以外の方）
①Windows 95 ②Windows 98 ③Windows Me ④Windows 2000 ⑤Windows NT ⑥Macintosh ⑦UNIX ⑧Linux ⑨その他

Q4　パソコン歴は？
①1年未満 ②1～2年 ③3～5年 ④6～10年 ⑤11年以上 ⑥なし

Q5　インターネット歴は？
①1年未満 ②1～2年 ③3～5年 ④6～10年 ⑤11年以上 ⑥なし

Q6　パソコンの主な使用目的は？（複数回答可）
①プログラミング ②ゲーム ③インターネット ④メール ⑤ミュージック ⑥学習 ⑦CG ⑧その他

Q7　これから買いたいと思うコンピュータ機器は？（複数回答可）
①パソコン本体 ②プリンタ ③スキャナ ④デジタルカメラ ⑤その他
※わかれば，商品名も記入してください

Q8　持っているゲーム機は？（複数回答可）
①プレイステーション（PSone含） ②プレイステーション2 ③ドリームキャスト ④ニンテンドウ64 ⑤ゲームボーイ ⑥ゲームボーイカラー ⑦ゲームボーイアドバンス ⑧ワンダースワン ⑨ワンダースワンカラー ⑩その他

Q9　これから買いたいと思うゲーム機は？（複数回答可）
①プレイステーション（PSone含） ②プレイステーション2 ③ドリームキャスト ④ニンテンドウ64 ⑤ゲームキューブ ⑥Xbox ⑦ゲームボーイカラー ⑧ワンダースワンカラー ⑨ゲームボーイアドバンス

※裏面へつづく→

■メモ欄…プログラム担当者・編集部へのご意見・ご感想など



第二十巻第十号通巻二三二号　昭和五十七年十二月六日第三種郵便物認可　二〇〇一年九月二十五日印刷　二〇〇一年十月一日発行

電波新聞社 〒141-8715 東京都品川区東五反田一−十一−十五 電話〇三（三四四五）六一一一

定価550円 本体524円 雑誌18361-10 T1118361100556

2001

わかる！動かせる！
プログラムが組める雑誌

マイコンBASIC Magazine

10

PS2でポリゴン・シューティング・ゲームの制作

http://www.basicmagazine.net/

電波新聞社